



**KAUNO TECHNOLOGIJOS UNIVERSITETAS
ELEKTROS IR ELEKTRONIKOS FAKULTETAS**

Martynas Ulinskas

IŠMANIOJO TERMOSTATO SUKŪRIMAS

Baigiamasis bakalauro projektas

Vadovas
Doc. dr. Vidas Raudonis

KAUNAS, 2015

KAUNO TECHNOLOGIJOS UNIVERSITETAS
ELEKTROS IR ELEKTRONIKOS FAKULTETAS
AUTOMATIKOS KATEDRA

IŠMANIOJO TERMOSTATO SUKŪRIMAS

Baigiamasis bakalauro projektas
Automatika ir valdymas (612H66001)

Vadovas

Doc. dr. Vidas Raudonis
2015 m. gegužės 28 d.

Recenzentas

Doc. dr. Tomas Tekorius
2015 m. gegužės 28 d.

Projektą atliko

Martynas Ulinskas
2015 m. gegužės 28 d.

KAUNAS, 2015

KAUNAS, 2015

TVIRTINU:

KTU Elektros ir elektronikos fakulteto

katedros vedėjas

201..... ..

BAKALAURO BAIGIAMOJO PROJEKTO UŽDUOTIS

Išduota studentui: Martynui Ulinskui Grupė *EVS-1*

1. Darbo tema:

Lietuvių kalba: Išmaniojo termostato sukūrimas

Anglų kalba: *Development of smart thermostat*

Patvirtinta 201__ m. _____ mėn. _____ d. dekanų potvarkiu Nr. _____

2. Darbo tikslas:

Sukurti ir iširti išmanųjį termostatą skirtą gyvenamojo pastato temperatūros valdymui pagal jo užimtumą. Sukurta temperatūros valdymo sistema turi atsižvelgti į pastate gyvenančių žmonių aktyvumą ir taip valdyti vidaus temperatūrą, kad būtų išlaikytas kuo geresnės komforto sąlygos.

3. Reikalavimai ir sąlygos:

Bakalauro projektas turi tenkinti bakalauro projektams keliamus reikalavimus, projektas turi būti parengtas Lietuvių kalba Microsoft Office teksto redagavimo programa. Kartu su dalykine bakalauro projekto dalimi studentas turi pateikti išeities kodus prieduose arba CD diske.

4. Projekto struktūra. Turinys konkretizuojamas kartu su vadovu, atsižvelgiant į BBP pobūdį, pateiktą Metodinių reikalavimų 14 ir 15 punktuose.

Literatūros apžvalga ir analizė:
Išanalizuoti literatūroje pateikiamą informaciją apie išmaniuosius termostatus, išsiaiškinti kaip jie yra realizuoti, kokiais pagrindiniais metodais ir algoritmais yra remiamasi.
Autoriaus teoriniai ir analiziniai tyrimai
Išanalizuoti internete prieinamą žmonių aktyvumo pasirinktame pastate duomenų bazę ir ją paruošti eksperimentiniams tyrimams.
Autoriaus eksperimentiniai tyrimai
Atlikti eksperimentinį tyrimą, nustatant kuri dirbtinio neuroninio tinklo struktūra yra geriausia tikslumo ir duomenų apdorojimo spartos atžvilgiais. Pateikti išvadas ir rekomendacijas išmaniojo termostato kūrimui.

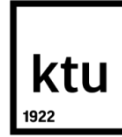
Užduotį gavau: Uliskas Martynas
(studento vardas, pavardė, parašas)

_____ (data)

Vadovas: Doc. Dr. Vidas Raudonis
(pareigos, vardas, pavardė, parašas)

_____ (data)

_____ (data)



KAUNO TECHNOLOGIJOS UNIVERSITETAS
ELEKTROS IR ELEKTRONIKOS FAKULTETAS

(Fakultetas)

Martynas Ulinskas

(Studento vardas, pavardė)

Automatika ir valdymas (612H66001)

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Pavadinimas“
AKADEMINIO SAŽININGUMO DEKLARACIJA

20 15 m. gegužės 28 d.
Kaunas

Patvirtinu, kad mano **Martyno Ulinsko** baigiamasis projektas tema „Išmaniojo termostato sukūrimas“ yra parašytas visiškai savarankiškai, o visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesažiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę įrašyti ranka)

(parašas)

Ulinskas, M. Išmaniojo termostato sukūrimas. Bakalauro baigiamasis projektas / vadovas doc. dr. Vidas Raudonis; Kauno technologijos universitetas, Elektros ir elektronikos fakultetas, Automatikos katedra.

Kaunas, 2015. 46 psl.

SANTRAUKA

Šiame darbe aprašomas išmanaus termostato sukūrimas. Termostatas kaupia pastate sumontuotų judesio jutiklių duomenis, juos apdoroja ir išskiria tris pastato užimtumo tipus. Sukaupus aštuonių savaitių duomenis, termostatas apmokomas taip, jog pagal einamąjį laiką prognozuotų pastato užimtumo tipą keletą valandų į priekį. Prognozavimui termostatas naudoja neuroninį tinklą. Kiekvienam užimtumo tipui būdinga konkreti temperatūra, kuriai esant gyventojai nepatiria šiluminio diskomforto. Termostatas, žinodamas galimus ateities užimtumo tipus ir jiems būdingas temperatūras, reguliuoja taip pastato šilumos sistemą, jog esant žemiausioms energijos sąnaudoms, gyventojai nepatiria šiluminio diskomforto. Atlikti tyrimai parodė, jog termostatui valdant pastato šilumą, 87% atvejų palaikoma norima temperatūra.

Reikšminiai žodžiai: pastato šildymas, termostatas, prognozavimas, elgsena, valdymas

Ulinskas, Martynas Development of smart thermostat . Final project of bachelor's / supervisor doc. dr. Vidas Raudonis; Kaunas University of Technology, Faculty of Electrical and Electronics Engineering, department of Automation.

Kaunas, 2015. 46 p.

SUMMARY

This paper describes Development of smart thermostat. Smart thermostat collects data from motion sensors and try to recognize three states of building occupancy. When it has building states of eight weeks thermostat trained to predict which state of building will be in the near future, by knowing current time. For prediction smart thermostat using artificial neural network. Each occupancy state has specific temperature because resident can't feel uncomfortable at specific time and occupancy state. Thermostat reduce energy use for house heating by controlling house heating knowing future house state of occupancy and specific temperature. Research show that using thermostat for controlling building heating 87% of the time temperature was at desirable level.

Keywords: house heating, thermostat, prediction, behavior, controlling

Turinys

| | |
|---|----|
| <i>SANTRUMPŲ IR ŽENKLŲ AIŠKINIMO ŽODYNAS</i> | 7 |
| <i>IVADAS</i> | 8 |
| <i>1. PASTATO ŠILUMOS REGULIAVIMO METODŲ APŽVALGA</i> | 9 |
| 1.1 <i>Sistemos, paremtos kaupiamų duomenų analize</i> | 9 |
| 1.1.1 <i>Išankstinio šildymo algoritmas</i> | 9 |
| 1.1.2 <i>Buvimo tikimybės algoritmas</i> | 11 |
| 1.1.3 <i>Išmanusis termostatas</i> | 12 |
| 1.1.4 <i>Išankstinio išjungimo sistema</i> | 13 |
| 1.1.5 <i>Neuroninis termostatas</i> | 14 |
| 1.2 <i>Sistema, paremta realių duomenų apdorojimu</i> | 16 |
| <i>2. SIŪLOMO METODO TEORINIS PAGRINDIMAS</i> | 18 |
| 2.1 <i>Dirbtinis neuroninis tinklas</i> | 18 |
| 2.1.1 <i>Matematinis neurono modelis</i> | 18 |
| 2.1.2 <i>Aktyvavimo funkcijos</i> | 18 |
| 2.1.3 <i>Vienasluoksnis neuroninis tinklas</i> | 20 |
| 2.1.4 <i>Daugiasluoksnis neuroninis tinklas</i> | 20 |
| 2.1.5 <i>Neuroninio tinklo mokymas</i> | 21 |
| 2.2 <i>PID reguliatorius</i> | 23 |
| <i>3. IŠMANIOJO TERMOSTATO PROGRAMINĖ REALIZACIJA</i> | 25 |
| 3.1 <i>Sistemos mokymui skirtų duomenų apdorojimas</i> | 25 |
| 3.2 <i>Pastato užimtumo timpų išskyrimas iš TSA duomenų</i> | 30 |
| 3.3 <i>Neuroninio tinklo mokymas</i> | 32 |
| 3.4 <i>Išmaniojo termostato veikimas</i> | 33 |
| <i>4. IŠMANIOJO TERMOSTATO VEIKIMO EKSPERIMENTINIAI TYRIMAI</i> | 37 |
| 4.1 <i>Duomenų apdorojimo tikslumas</i> | 37 |
| 4.2 <i>Neuroninio tinklo prognozavimo tikslumas</i> | 38 |
| 4.3 <i>Išmaniojo termostato veikimo tikslumas</i> | 44 |
| <i>REZULTATAI IR IŠVADOS</i> | 45 |
| <i>INFORMACIJOS ŠALTINIŲ SĄRAŠAS</i> | 46 |
| <i>PRIEDAI</i> | 47 |
| <i>Priedas 1. M matricos formavimo Matlab išeities kodas</i> | 47 |
| <i>Priedas 2. Duomenų interpoliavimo Matlab išeities kodas</i> | 47 |
| <i>Priedas 3. Duomenų išlyginimo Matlab išeities kodas</i> | 48 |
| <i>Priedas 4. N ir T imčių sudarymo Matlab išeities kodas</i> | 48 |
| <i>Priedas 5. Išmaniojo termostato Matlab išeities kodas</i> | 49 |

SANTRUMPŲ IR ŽENKLŲ AIŠKINIMO ŽODYNAS

| | | |
|------|--|---|
| PIR | (angl. <i>Passive infrared sensor</i>) | Pasyvusis infraraudonųjų spindulių jutiklis |
| RFID | (angl. <i>Radio frequency identification</i>) | Radijo dažnio nustatymo jutikliai |
| GPS | (angl. <i>Global positioning system</i>) | Globalinė padėties nustatymo sistema |
| HMM | (angl. <i>Hidden Markov model</i>) | Paslėpti Markovo modeliai |
| PP | (angl. <i>Presence probabilities</i>) | Buvimo tikimybė |
| ANN | (angl. <i>Artificial neural network</i>) | Dirbtinis neuroninis tinklas |
| TSA | (angl. <i>Thermally smart apartment</i>) | Tulum išmanus būstas |
| IT | | Išmanus termostatas |
| IIS | | Išankstinio išjungimo sistema |
| NT | | Neuroninis termostatas |

IVADAS

Pasaulinis energijos suvartojimas per šešiasdešimt metų padidėjo beveik tris kartus. Remiantis Energijos Informacijos Administracijos [1], energijos suvartojimas iki 2040 metų vidutiniškai didės 2% per metus. Galima teigti, jog penktadalis visos energijos yra reikalinga namų ūkio poreikiams [2].

Pastato šildymas sudaro pagrindines energijos sąnaudas. Šildymui tenka iki 40% namų ūkių energijos poreikių [3]. Keičiant pastato šildymą, atsižvelgiant į tai, ar gyventojai miega, ar yra išvykę, galima sutaupyti nuo 10% iki 30% energijos poreikių pastato šildymui [4]. Tačiau tai realizuoti yra gana sudėtinga, nes retas vartotojas reguliuoja termostatą keletą kartų per dieną, o įdiegęs programuojamą termostatą moka tinkamai jį sukonfigūruoti.

Pastatui šildyti turi būti rastas kompromisas tarp energijos sąnaudų ir šilumos komforto. Naudojant programuojamą termostatą, tam tikros paros metu galima nustatyti norimą temperatūrą ir taip sumažinti energijos sąnaudas, tačiau, jei vartotojas suklysta programuodamas termostatą, yra sumažinamos komforto sąlygos. Kitas būdas – įdiegti sistemą, kuri pati nustatytų, kada ir kokios temperatūros reikia. Jungtinėse Amerikos Valstijose, Kolorade, name buvo įdiegta prisitaikanti sistema *Ache*, paremta neuroniniu tinklu, kurios tikslas buvo sumažinti energijos poreikius. Sistema, saugodama duomenis apie vartojamą šviesą, termostato reguliavimą, langų ir durų padėtį ir judesio jutiklių duomenis, prognozuodavo vartotojo elgseną ir atitinkamai valdė pastato šildymą ir apšvietimą [5]. Europoje plėtojame *ERGDOM* projekte siūloma, reguliuojant pastato šilumą užtikrinti malonią temperatūrą. Ši sistema, apdorodama istorinius judesio jutiklių duomenis ir vartotojo nustatytas temperatūras, prognozavo reikalinga pastato šilumą konkrečiu laiku [6]. Apibendrinus *ERGDOM* ir *Ache* sistemas galima jas pavadinti išmaniaisiais termostatais.

Išmaniojo termostato nereikia programiškai nustatinėti, kada ir kokia temperatūra pageidaujama, tačiau tuo pat metu jis užtikrina aukščiausias šilumos komforto sąlygas. Pirmą išmanų termostatus namų vartotojams pasiūlė kompanija *Nest labs*. Jų išmanusis termostatas, naudodamas temperatūros, judesio, šviesos, drėgmes jutiklius ir orų duomenis, gaunamus WiFi ryšiu, sukuria kiekvienos savaitės dienos šildymo vedinimo grafiką ir pagal jį reguliuoja pastato šilumą. Kitų kompanijų siūlomi termostatai gali būti valdomi balsu, sekti telefono, su kuriuo susietas, buvimo vietą ir pagal ją nustatyti ar gyventojas namie, ar išvykę.

Darbe apžvelgiamos dažniausios pastato šildymo reguliavimo sistemos, taikomi metodai. Jais remiantis suprojektuojama automatinė pastato šildymo sistema, kuri apdorodama istoriniu duomenis prognozuoja gyventojų veiklas ir parenka atitinkamą temperatūrą.

1. PASTATO ŠILUMOS REGULIAVIMO METODŲ APŽVALGA

Pagrindinis visų šilumos reguliavimo sistemų tikslas yra palaikyti malonę pastato temperatūrą, naudojant kuo mažiau energijos kiekio reikalingo pastato šildymui. Energijos sąnaudos šildymui didėja didinant norimos temperatūros lygį ir mažėja – mažinant. Mažinant pastato temperatūrą mažėja ne tik energijos sąnaudos, bet ir gyventojų pasitenkinimas šilumine aplinka, todėl sistema visų pirma turi išskirti pastato užimtumo tipus, kuriems būdinga žemesnė temperatūra. Antras sistemos veiksmas yra sąlygojamas pastato šiluminės inercijos, dėl kurios staigiai padidinti ar sumažinti temperatūros negalima. Norint, kad patalpos būtų prišildytos laiku, šildymo sistema turi žinoti pastato ateities užimtumo tipą. Laikas, kurį sistema turi numatyti į priekį, priklauso nuo to, kiek laipsnių per valandą sistema gali pakelti patalpų temperatūrą.

Pastato užimtumams išskirti naudojami pasyvieji infraraudonųjų spindulių jutikliai (angl. *passive infrared sensors*) (PIR), radijo dažnio nustatymo jutikliai (angl. *radio frequency identification*) (RFID) ar vaizdo kameros, kurie sumontuojami pastate. Gyventojų buvimo vietai nustatyti taip pat naudojamos padėties nustatymo sistemos (angl. *global positioning system*) (GPS).

Pastato užimtumo tipams prognozuoti naudojami neuroniniai tinklai, paslėpti Markovo modeliai (angl. *Hidden Markov Model*) (HMM) [5], [7] ar kitokie algoritmai [4],[8],[9]. Visas sistemas galima suskirstyti į dvi kategorijas pagal duomenų, naudojamų prognozuoti pastato užimtumui, tipą. Konkrečių jutiklių duomenys gali būti kaupiami ir juos analizuojant prognozuojamas galimas pastato užimtumas, arba apdorojami realūs jutiklių duomenys.

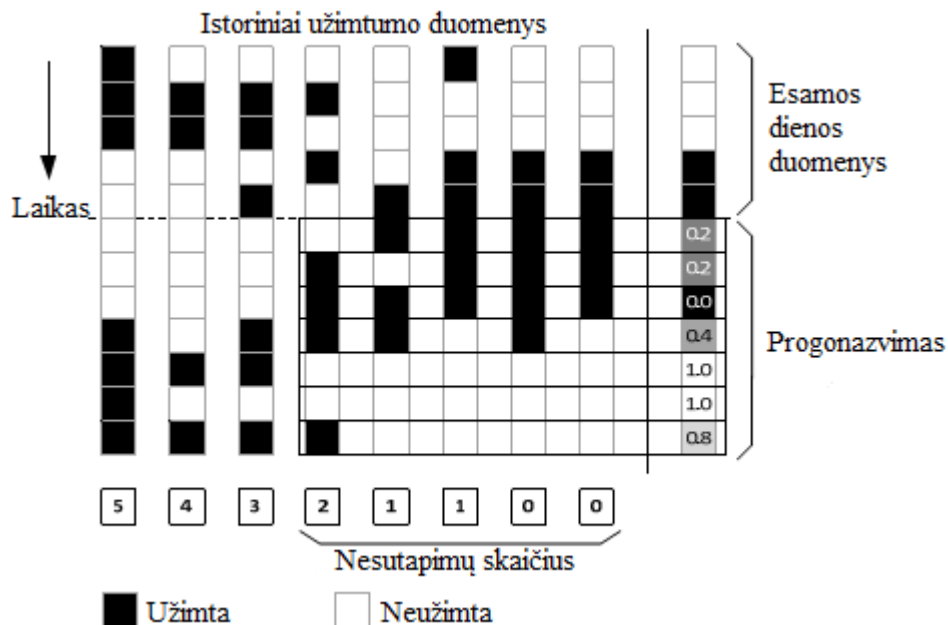
1.1 *Sistemos, paremtos kaupiamų duomenų analize*

Šioms sistemoms būdinga tai, kad jos prognozuoja užimtumą, pasitelkdamos tik kaupiamus jutiklių duomenis. Paprastai tam, jog būtų galima prognozuoti užimtumą, būna saugojami kelių savaičių ar mėnesių duomenys.

1.1.1 *Išankstinio šildymo algoritmas*

James Scott išankstinio šildymo algoritmas [4] yra puikus sukauptų duomenų analizavimu paremtas algoritmo pavyzdys. Algoritmas skiria trijų rūšių temperatūras – miego, tuščios ir užimtos patalpos. Pirmiausiai nustatoma, ar patalpos užimtos, ar tuščios. Patalpoms esant užimtoms naudojama užimtos patalpos temperatūra, o naktį yra palaikoma miego temperatūra. Patalpoms esant tuščioms sistema apdorojusi istorinius duomenis, prognozuoja, kada jos vėl bus užimtos

Kiekvienos dienos užimtumo duomenys pateikiami dvejetainiu vektoriumi, kurio kiekvienas elementas atitinka 15 min intervalą (1.1 pav.). Jeigu fiksuojamas patalpos užimtumas, tai vektoriaus elementas yra lygus 1, kitu atveju 0. Prognozuojant užimtumą ieškoma K panašiausių vektorių į esamos dienos duomenis. Remiantis James Scott atliktais eksperimentais [4], norint užtikrinti aukštą tikslumo lygį, K turėtų būti ne mažesnis kaip 5.



1.1 pav. James Scott užimtumo prognozavimo algoritmas [4]

Norint padidinti prognozės patikimumą yra atskiriamos darbo dienos ir savaitgaliai. Darbo dienų prognozė skaičiuojama tik iš praėjusių darbo dienų, o savaitgaliai iš savaitgalių. Kitas sistemos patikimumą lemiantis principas yra tai, jog prie kiekvienos dienos užimtumo vektoriaus pradžios ir pabaigos yra pridedami 4 valandų praėjusios ir ateinančios dienos užimtumo duomenys. Kas padeda tiksliau prognozuoti paros pradžios ir pabaigos įvykius. Šios sistemos privalumas yra tai, jog užimtumas prognozuojamas kaip tikimybė. Todėl yra pasitelkiama slenkstinė reikšmė α , nuo kurios priklauso, kada bus laikoma, jog erdvė yra užimta. Siekiant, jog sistema funkcionuotų patikimai, nustatomas didesnis α . Priešingu atveju, norint jog patalpos būtų šiltesnės, α parenkamas mažesnis.

Sistema buvo įdiegta penkiuose namuose, kuriuose buvo vertinami tokie aspektai:

- šiluminio diskomforto laiko sumažėjimas;
- dujų, skirtų pastato šildymui, sumažėjimas.

1.1 lentelėje sistema lyginama su programuojamu ir pastoviu šildymu. Buvo pastebėta, kad kai kuriose pastato patalpose, kurios dažniausiai būna užimtos nesistemiškai ir trumpai, tikslumas žemas. Buvo nuspręsta, jog tokios patalpos bus laikomos visados neužimtomis ir jas sistema šildys intuityviai.

1.1 lentelė. James Scott sistemos lyginimas su programuojamu ir pastoviu šildymu [4].

| | Namas 1 | Namas 2 | Namas 3 | Namas 4 | Namas 5 |
|---|---------|---------|---------|---------|---------|
| Vidutinis namo užimtumas | 89% | 69% | 67% | 73% | 68% |
| Šiluminio diskomforto laiko sumažėjimas lyginant su programuojamu šildymu | 88% | 84% | 92% | 38% | 60% |
| Dujų vartojimo sumažėjimas lyginant su programuojamu šildymu | -5% | -1% | 2% | 18% | 8% |
| Dujų vartojimo sumažėjimas lyginant su pastoviu šildymu | 3% | 17% | 10% | 27% | 35% |

Iš James Scott tyrimo rezultatų matyti, jog sistema su užimtumo prognozavimu ne visuomet pranašesnė už programuojamą šildymą ir dujų suvartojimas gali padidėti iki 5%, tai priklauso nuo sugebėjimo tinkamai parinkti laiką ir norimą temperatūrą programuojamai šildymo sistemai. Tačiau palyginus su pastoviu namo šildymu, dujų suvartojimą galima sumažinti iki 35%.

1.1.2 Buvimo tikimybės algoritmas

J. Krumm ir B. Brush aprašė buvimo tikimybės (angl. *Presence Probabilities*) (PP) algoritmą [8], kuris pagrįstas GPS duomenimis. Pastato užimtumas yra nustatomas naudojantis GPS siųstuvais, kuriuos turi visi šeimos nariai. Pastatas laikomas užimtu, jeigu nustatoma, jog vienas iš GPS siųstuvų yra arčiau negu 100 metrų. Siųstuvai registravo koordinates kas penkias sekundes. PP algoritmas, naudodamas GPS duomenis apskaičiuoja tikimybę $p_{neužimta}$. Tikimybė gaunama GPS duomenų sumą $n_{neužimta}$, kai pastatas neužimtas, padalinus iš visų nuskaitymų skaičiaus atitinkamame 30 minučių intervale (1). Taip gaunamas vektorius atspindintis pastato neužimtumo tikimybę 30 minučių intervalu.

$$p = \frac{n_{neužimta}}{n_{užimata} + n_{neužimta}} \quad (1.1)$$

Visos savaitės neužimtumo tikimybės pavaizduojamos nežinomos formos vektoriumi $p_{savaitės} = (p_1, \dots, p_{336})^T$, čia vienas vektoriaus elementas atitinka trisdešimt minučių intervalą. Vektoriaus pirmasis elementas p_1 atitinka pirmas 30 pirmadienio minučių, o p_{336} paskutines 30 sekmadienio minučių.

Atsižvelgiant į tai, jog darbo dienomis yra pastovesnė dienotvarkė, negu savaitgaliais, apskaičiuojamas bendras darbo dienų neužimtumo tikimybės vektorius $p_{bendra\ d.d.} = (p'_1, \dots, p'_{48})^T$. Vektorių sudaro 48 elementai, tai atitinka 24 valandų, bendros darbo dienos, trisdešimties minučių periodą. Vektoriaus $p_{bendra\ d.d.}$ svarbą, bendrai nebuvimo tikimybei $p_{neužimta}$, galima keisti reguliavimo faktoriumi λ . Darbo dienų neužimtumo tikimybė

apskaičiuojama sudėjus $p_{savaitės}$ ir $p_{bendra\ d.d.}$, savaitgalių užimtumo tikimybė yra apskaičiuojama remiantis $p_{savaitės}$ vektoriaus duomenimis.

Prieš pradėdant tyrimus, eksperimente sutikusių dalyvauti žmonių buvo paprašyta sudaryti buvimo namuose grafiką. Gyventojų grafikai buvo palyginti su GPS duomenimis (1.2 lentelė), laikant, jog GPS duomenys yra teisingi. 24% laiko, kurį sistema fiksavo, jog gyventojai yra namuose, gyventojai teigė, kad jie bus išvykę. O 68% laiko, kai fiksavo išvykimą, gyventojai manė, kad bus namuose. Galima daryti išvadą, jog gyventojai manė, jog bus daugiau laiko namuose. Gyventojai sunkiai prognozuoja kada jie bus namuose ir kada ne, todėl programuojant termostata, pagal vartotojo nustatytą grafiką, vyksta perteklinis patalpų šildymas, kuris lemia didėjančius energetinius sąnaudas.

1.2 lentelė. GPS duomenų ir gyventojų prognozės palyginimas [8].

| | | Gyventojų prognozė | |
|--------------|---------|--------------------|--------|
| | | Namuose | išvykę |
| GPS duomenys | Namuose | 76% | 24% |
| | Išvykę | 68% | 32% |

Neužimtumo tikimybinės lentelės naudojimo privalumai:

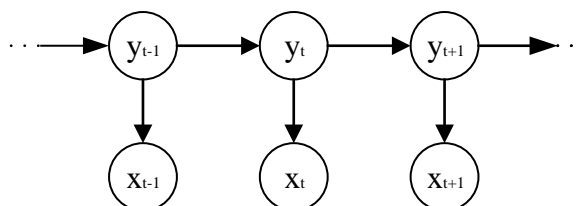
- lentelė sudaryta remiantis realiu gyventojų savaitės grafiku;
- neužimtumo išreiškimas tikimybe parodo, jog žmonių veikla nėra iki galo prognozuojama.

1.1.3 Išmanusis termostatas

Jiakang Lu [7] aprašė namo šildymo sistemą, kuri valdoma išmanaus termostato (IT). Termostatui nereikia vaizdo kamerų ar nešiojamų siūstuvų. Jis naudoja parastus judesio, ir durų jutiklius. Kambariuose sumontuoti PIR jutikliai, o ant lauko durų magnetiniai jutikliai. Remiantis jutiklių duomenimis sistema veikia trimis energijos taupymo būdais:

- sistema, naudodama jutiklių duomenis, greičiau nei per minutę atpažįsta, kad patalpos tuščios, jose miegama ar aktyviai etapais;
- naudodama istorinius jutiklių duomenis kartu su momentiniais duomenimis sistema prognozuoja, kada reikia įjungti ir išjungti šildymo sistemą,
- sistema mažina energijos poreikį išjungdama šildymo sistemą, kai patalpos nėra užimtos.

Prognozavimui termostatas naudoja HMM (1.2 pav.),



1.2 pav. HMM [7]

kur y_t yra pastato užimtumo tipas, x_t yra įėjimo vektorius, kurį sudaro trijų tipų jutiklių duomenys:

- keturių valandų jutiklių reikšmės;
- bendras visų jutiklių suveikimų skaičius, per laiko intervalą dT ;
- durų padėtys, dvejetainė forma.

Pirmi duomenys nurodo istorines jutiklių reikšmes ir padeda HMM nustatyti esamą užimtumą. vėlesni duomenys nustato užimtumo lygį, paskutiniai duomenys padeda nustatyti, kur yra atidarytos ir uždarytos durys. Pastato užimtumas y_t yra apskaičiuojama 5 minučių intervalui.

Žinant užimtumą, sistema sprendžia išjungti ar įjungti šildymo sistemą. Pastato šildymas išjungiamas, kai gyventojai išvyksta. Tačiau pastato šildymo įjungimo laikas priklauso nuo esamos pastato vidaus temperatūros ir prognozuojamos gyventojų grįžimo laiko. Pradėjus šildyti per anksti, dėl aukštos temperatūros palaikymo, išauga energijos poreikis, tačiau pavėluotas šildymo sistemos įjungimas lemia, jog šildymo sistema, norit greitai pakelti pastato temperatūrą, bus perjungta į neefektyvų darbo režimą. IT nustato optimalų pastato prišildymo laiką τ , kuris parenkamas siekiant sumažinti energijos poreikius šildymui iki minimumo. Nustatant prišildymo laiką τ , sistema remiasi praėjusių dienų duomenimis ir nustato visus galimus grįžimo laikus $a = [a_0, a_1, \dots, a_n]$. Visiems galimiems prišildymo laikas $t_i \in [\max(a), \min(a)]$ sistema apskaičiuoja energijos poreikį $w_i(t_i)$, kuris būtų reikalingas prišildyti patalpas per laiką t_i . Prišildymo laikas τ nustatomas t_k , prie kurio energijos sąnaudos w_k yra mažiausios.

Daugiausiai energijos šildymui sutaupoma, kai nesant šilumos poreikiui mažinama vidaus temperatūra. Sutaupomos šilumos kiekis priklauso nuo temperatūros kritimo ΔT ir laiko $t_{\Delta T}$ kurį pastato vidaus temperatūra laiko ΔT laipsnių žemesnė nei pageidautina. Labiausiai sumažinti temperatūra galima, kai pastatas nėra užimtas., tačiau reikia žinoti kiek galima sumažinti temperatūrą, jog sistema spėtų pakelti temperatūrą iki greičiausio tikėtino grįžimo laiko $\min(a)$.

Tyrimo rezultatai parodė, jog namuose, kuruose yra pastovūs užimtumo grafikai energijos kaštai sumažėja iki 38,6%, o pastatai, kurie reguliariai užimti visą dieną energijos kaštai sumažėja 17,4%. Tuo pačiu metu reikiama temperatūra nebuvo užtikrinama vidutiniškai 45 minutes per dieną.

1.1.4 Išankstinio išjungimo sistema

C. Ellis su bendraautoriais aptarė išankstinio išjungimo sistemą (IIS) [9]. Mokslininkai rėmėsi pastato šiluminę inercija. Šiluminė inercija yra svarbi pastato charakteristika, kuri tam tikrą laiką tarpą, po šildymo sistemos išjungimo, patalpose palaiko temperatūrą. ISS prognozuoja, kada gyventojai paliks patalpas ir dar prieš gyventojams išvykstant, žinant jog dėl šiluminės inercijos pastatas išlaikys komforto temperatūrą n minučių, išjungia šildymą.

Išankstinio šildymo sistema taupo energiją, išjungdama šildymą. Tai daroma remiantis, jog prieš išvykstant gyventojams reikalinga žemesnė temperatūra, nes jie būna apsirengę, ar aktyvūs. Buvo pasirinkta, jog prieš išvykstant patalpos gali atvėsti 1°C.

Mokslininkai pasinaudojo PP algoritmu [8], tačiau vietoj prognozuojamos nebuvimo tikimybės, algoritmas prognozavo diskretinius išvykimo įvykius. Išvykimas buvo fiksuojamas, ieškant reikšmingo buvimo tikimybės kritimo. Kiekvienai dienai buvo apskaičiuotas standartinis nuokrypis tarp paeilui einančių laikų intervalų. Žinant standartinį nuokrypį buvo ieškoma laiko intervalų, kurių buvimo tikimybės sumažėjimas yra bent 2,5 karto didesnis už standartinį nuokrypį. Siekiant sumažinti netikslius spėjimus, kurie labiau tikėtini antroje dienos pusėje, buvo apribotas galimas išvykimo laikas tarp 7:00 ir 14:00.

Skaičiuojant algoritmo efektyvumą, buvo lyginamas šildymo sistemos be ISS ir su ISS dujų suvartojimo kiekis ir kiek dienų sistema tiksliai prognozavo išvykimo laiką. Algoritmas buvo testuojamas su penkių namų užimtumo duomenimis. Išvykimo duomenys buvo prognozuojami 30 – 84% tikslumu, o dujų sąnaudos sumažėjo 1 – 8 %. Šis algoritmas gali būti kaip papildantis tokias sistemas kaip išankstinio šildymo, išmanaus termostato ar kitas.

1.1.5 Neuroninis termostatas

C. Mozer optimaliai namo šildymo sistemai pasiūlė neuroninį termostatą (NT). NT, remdamasis pastato šiluminės varžos ir užimtumo prognozavimo medeliais, ieško mažiausių energijos sąnaudų esant pakankamam šiluminiam komfortui. Vidutiniai tikėtini kaštai gali būti išreiškiami:

$$J = \lim_{K \rightarrow \infty} \sum_{t=t_0+1}^{t_0+K} [e(u_t) + m(x_t)], \quad (1.2)$$

čia t yra nepersidengiantys vienodo ilgio laiko intervalai remiantis, jog aplinkos kintamieji yra konstantos. t_0 yra einamojo intervalo pabaiga, u_t yra, konkretaus laiko intervalo t , sistemos sprendimas įjungti ar išjungti šildymą. $e(u)$ yra energetinių kaštų funkcija, kuri priklauso nuo sistemos sprendimo u . x_t yra aplinkos būseną laiko intervale t , kurią sudaro vidaus oro temperatūra ir užimtumo tipas name. $m(x)$ yra funkcija išreiškianti šiluminį diskomfortą esant aplinkos sąlygoms x . Kad būtų galima sudėti šiluminį diskomfortą ir energetinius kaštus, jiems suteikiama pinigine išraiška.

Energijos kaštai priklauso nuo šilumos sistemos veikimo, tačiau šiluminio diskomforto kaštai priklauso nuo namo užimtumo $o(t)$ ir vidaus temperatūros $h_u(t)$:

$$m_u(o(t), h_u(t)) = p[o(t) = 1] m(1, h_u(t)) + p[o(t) = 0] m(0, h_u(t)). \quad (1.3)$$

Iš pastato šiluminio modelio gaunama energetiniai kaštai e ir vidaus temperatūra h_u :

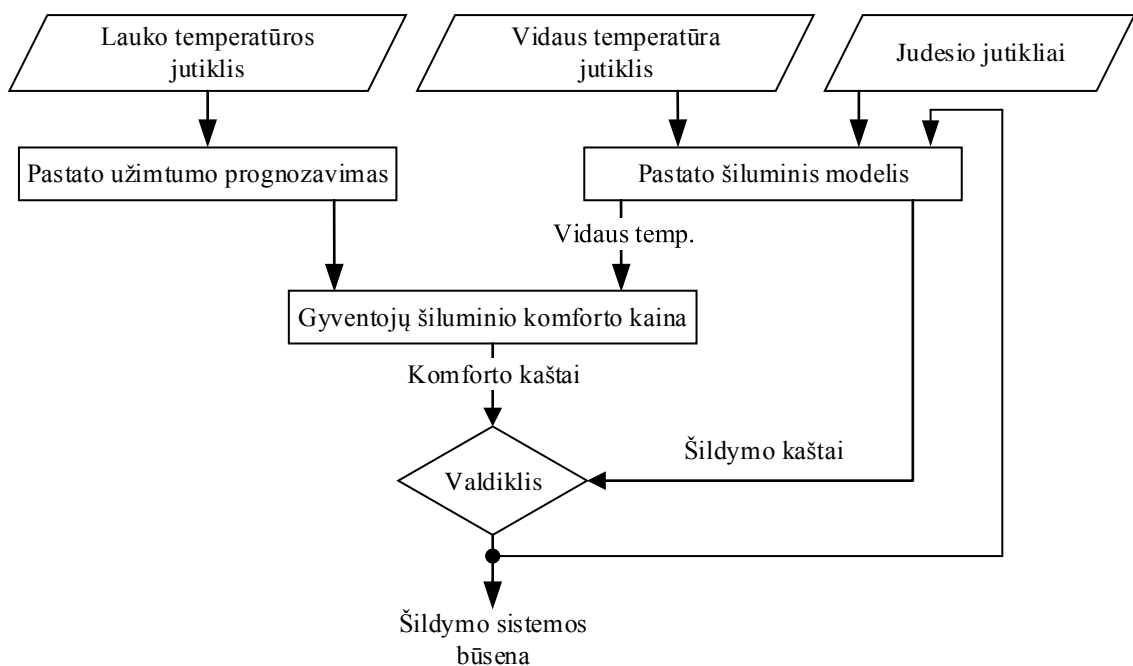
$$h_u(t) = h_u(t-1)e^{-\frac{60\delta}{RC}} + (RQu(t) + g)\left(1 - e^{-\frac{60\delta}{RC}}\right), \quad (1.4)$$

kur R – pastato sienų temperatūrinė varža, C – temperatūrinė talpa, Q – šilumos kiekis patenkantis į patalpas. Gyventojų šiluminio komforto modelis apskaičiuoja komforto kainą m :

$$m(o, h) = o\alpha \frac{\delta}{24 \cdot 60} * \frac{\max(o, |\lambda - h| - \epsilon)^2}{25}, \quad (1.5)$$

čia o – pastato užimtumo tipas, $\alpha = \gamma\rho$, kur α – diskomforto kaina doleriais, ρ – prarastas darbo našumas valandomis, dirbant 5°C žemesnėje, negu numatyta temperatūroje, γ – valandinis užmokestis. Ketvirtoji modelio dalis išreiškia 1 diskomforto vienetą, esant 5°C temperatūrų skirtumui ir nejautos zonai nuo $\lambda - \epsilon$ iki $\lambda + \epsilon$. λ – šiluminio komforto temperatūra.

Neuroninio termostato struktūra pavaizduota 1.3 paveiksle. Svarbiausia termostato dalis yra valdiklis, kuri kas δ minučių gali išjungti arba įjungti šildymą. Kiekvienam laiko intervalui valdiklis apžvelgia visus galimus veiksmus $n\delta$ minučių į ateitį ir pasirenka tokią veiksmų seką, kuri užtikrintų mažiausius kaštus J .



1.3 pav. Neuroninio termostato struktūra [10].

Gyventojų užimtumo modelis naudodamas neuroninį tinklą prognozuoja p reikšmę. Užimtumas prognozuojamas manant, jog gyventojai laikosi pastovios dienotvarkės. Laikas yra suskirstomas δ minučių intervalais. Neuroninis tinklo įėjimai yra:

- einamosios dienos laikas;
- savaitės diena;
- vidutinis 10, 20 ir 30 minučių pastarųjų trijų dienų ir keturių, tos pačios savaitės, dienų laikai, kurį patalpa buvo užimta nuo esamo dienos laiko;
- einamosios dienos vidutinis 60, 180, 360 paskutinių minučių laikas, kurį patalpa buvo užimta.

Tinklas pasirenkamas trijų sluoksnių, paslėptų neuronų skaičius parenkamas pagal tenkinamą paklaidą [10].

1.2 Sistema, paremta realių duomenų apdorojimu

Visi aptarti metodai prognozavo, kada įjungti ir išjungti šildymą pagal istorinius duomenis. Kitas būdas valdyti pasato šilumos sistemą fiksuojant duomenis realiu laiku.

Mokslininkai [11] pasiūlė pastato šilumos sistemą valdyti priklausomai nuo vartotojo galimo grįžimo laiko, kuris apskaičiuojamas pagal vartotojo mobilaus telefono vietą. Termostatas pastoviai sąveikauja su GPS siųstuvu, esančiu telefone, kuris nurodo asmens buvimo koordinatės. Sistema iš turimų koordinatė apskaičiuoja greičiausią grįžimo laiką t_g , kurį vartotojas užtruktų grįžti iš esamos vietos. Žinant lauko ir vidaus temperatūras, pastato šildymo charakteristiką, sistema lygina šildymo laiką t_s su grįžimo laiku t_g . Jeigu $t_s \geq t_g$ sistema įjungia šildymo sistemą. Šios sistemos privalumai:

- nereikalingas termostato programavimas;
- prisitaiko prie nereguliaraus vartotojo grafiko;
- užtikrina, jog vartotojui grįžus, bus komfortabili temperatūra;
- sistemai nereikia montuoti papildomų jutiklių.

Sistema kas vieną minutę nuskaito lauko ir vidaus temperatūras, kiekvieno pastato gyventojų telefono ilgumos ir platumos koordinatės ir namo šildymo/vėsinimo lenteles. Šildymo lentelės kiekvienam namui yra apskaičiuojamos atskirai. Lentelėje nurodoma laikas, kurio reikia pakelti pastato temperatūrą 1°C nuo pradinės temperatūros T_p , esant T_l lauko temperatūrai.

Gavus ilgumos /platumos koordinatė reikšmes termostatas jas siunčia į žiniatinklio tarnybą *MapQuest*, kad apskaičiuotų grįžimo namo laiką. *MapQuest* tinklo mazgas pagal kelių žemėlapi, nustato atstumą ir atsižvelgiant į leistiną maksimalų greitį apskaičiuoja grįžimo laiką. Tačiau sistema neįvertina eismo sąlygų, dėl kurių gali pailgėti grįžimo laikas. Išorės temperatūra gaunama tiesiogiai prisijungus prie orų duomenų bazės.

Sistemos efektyvumas buvo lyginamas su rankiniu būdu valdomu termostatu, ir termostatu, kurį suprogramavo vartotojas. Iš keturiuose namuose atlikto tyrimo matyti, jog efektyviausiai veikė programuojamas termostatas sutaupydamas 13,5 – 19,4% energijos sąnaudų. Rankinis termostatas sutaupė 12,6 – 18,0%. GPS termostatas sutaupė 0,3 – 7,1% energijos sąnaudų šilumai.

Naudojant GPS duomenis galima pagerinti GPS termostato efektyvumą, sukuriant sistemą, kuri užfiksuoja, jog gyventojas išvyksta iš namų ir nebejuda. Užfiksavus, kad gyventojas nebejuda, sistema į vartotojo telefoną pateikia keletą klausimų:

- Nustatyti žemesnę temperatūrą. Vartotojas sutinka, jog jam grįžus namo pastato temperatūra būtų nežymiai mažesnė, negu pageidautina, tačiau toliau šildoma, kol pasiekiami komforto temperatūra.
- Nustatyti tikslų grįžimo laiką. Jeigu vartotojas nusprendžia grįžti atitinkamu laiku, termostatas tiesiog ignoruos GPS duomenis, tuomet sistema apskaičiuos mažiausią išvykimo temperatūrą ir reikiamą šildymo pradžią, jog grįžus gyventojams būtų pasiekta norima temperatūra.
- Nustatyti žemesnę temperatūrą tiksliu laiku. Kombinuojant tikslų šildymo laika ir žemesnę temperatūrą pasiekiami mažiausi energijos kaštai.

Nurodoma, kad termostatui naudojant GPS duomenis ir pateikiant vartotojui nesudėtingus klausimus, galima sutaupyti iki 19,21% energijos sąnaudų [11].

2. SIŪLOMO METODO TEORINIS PAGRINDIMAS

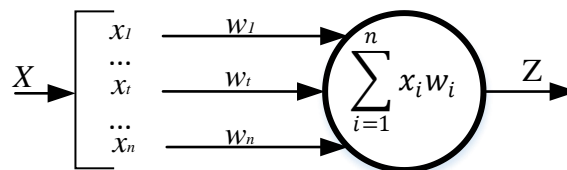
2.1 Dirbtinis neuroninis tinklas

Mus supančioje aplinkoje egzistuoja begalės užduočių, kurias kompiuteris paprastai išsprendžia, tačiau žmogui tai padaryti gana sudėtinga, pavyzdžiui: ištraukti kvadratinę šaknį iš šešiaženkliai skaičiaus. Kitą vertus, yra užduočių, kurios paprastos žmogui, bet sudėtingos kompiuteriui. Kiekvienas žmogus pamatęs nuotrauką lengvai atskirs šuniuką nuo kačiuko, ar supras, kas parašyta skirtingu raštu. Buvo sukurtas matematinis skaičiavimo modelis, paremtas žmogaus smegenų veikimo principu, kuris suteikė galimybę kompiuteriui atlikti tokias funkcijas.

Pirmą dirbtinio neuroninio tinklo (angl. artificial neural network) (ANN) medelį 1943 metais suformavo W.S. McCulloch ir W. Pitts. Pagal jų pasiūlytą modelį smegenys yra sudaryti iš daug neuronų, kurie turi vienodą struktūrą. Neuroną galima įsivaizduoti kaip slenkstinę funkciją – tik viršijus slenkstinę vertę, suformuojamas išėjimo signalas.

2.1.1 Matematinis neurono modelis

Dirbtinis neuronas imituota biologinio neurono veikimo principą. Neurono įėjimas yra X vektorius, kurį mes gauname kaip pradinius duomenis arba, kaip kitų neuronų išėjimus. Kiekviena X vektoriaus komponentė x_i dauginama iš svorio reikšmės w_i . Visų sandaugų suma lemia neurono aktyvavimo lygį. Dirbtinio neurono medelis parodytas 2.1 paveiksle. Įėjimo vektorius $X = (x_1, x_2, \dots, x_n)$ dauginamas iš atitinkamo įėjimo svorio koeficiento $W(w_1, w_2, \dots, w_n)$ ir visos gaunamos sandaugos sudedamos.

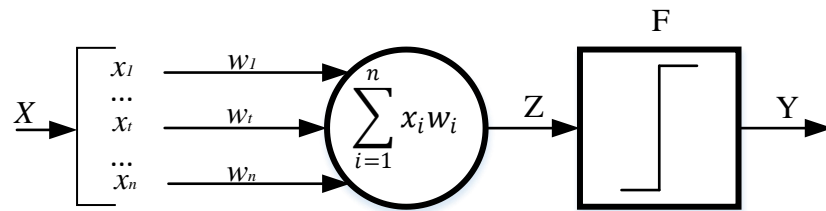


2.1 pav. Dirbtinio neurono modelis

Įėjimo vektorius $X = (x_1, x_2, \dots, x_n)$ dauginamas iš atitinkamo įėjimo svorio koeficiento $W(w_1, w_2, \dots, w_n)$ ir visos gaunamos sandaugos sudedamos.

2.1.2 Aktyvavimo funkcijos

Pagal sužadavimo signalą Z , naudojant aktyvavimo funkciją F , gaunamas neurono išėjimo signalas Y (2.2 pav.).



2.2 pav. Dirbtinio neurono modelis su aktyvavimo funkcija

Dažniausiai naudojami aktyvavimo funkcijos:

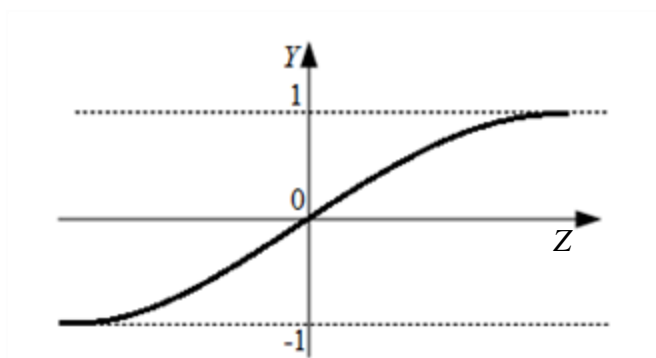
- slenkstinė funkcija (2.1) (2.3 pav., b),
- sigmoidinė funkcija (2.2) (2.4 pav., a),
- hiperbolinis tangentas (2.3) (2.5 pav., c),
- Tiesinė funkcija (2.4) (2.6 pav., d).

$$Y = F(Z) = \begin{cases} 1, & \text{jeigu } Z \geq k \\ 0, & \text{jeigu } Z < k \end{cases} \quad (2.1)$$

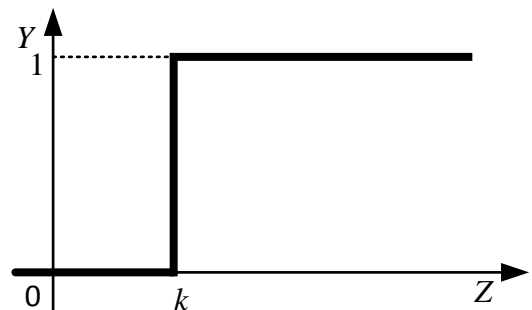
$$Y = F(Z) = \frac{1}{1+e^{-Z}} \quad (2.2)$$

$$Y = F(Z) = \frac{e^Z - e^{-Z}}{e^Z + e^{-Z}} \quad (2.3)$$

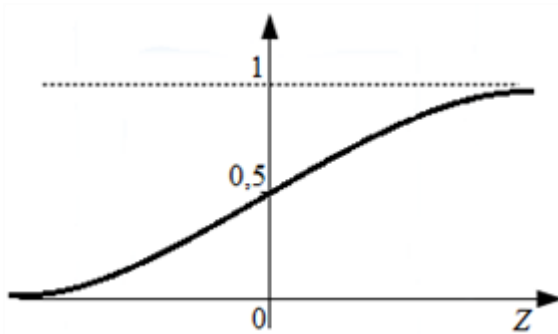
$$Y = F(Z) = kZ \quad (2.4)$$



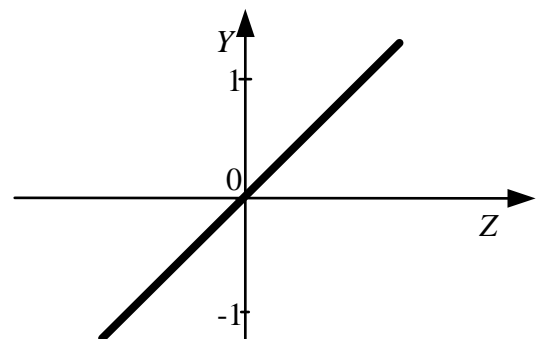
a)



b)



c)

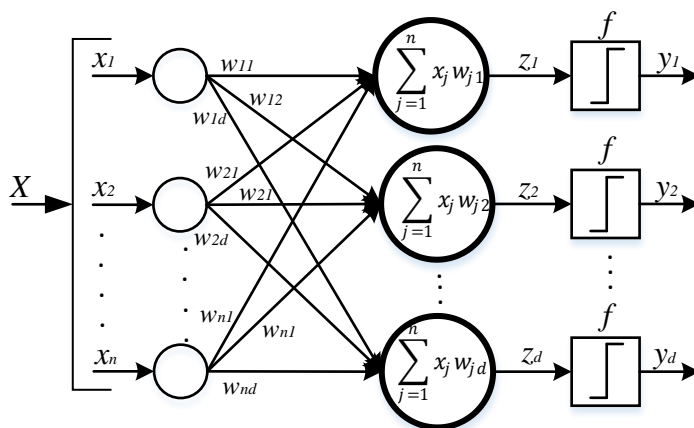


d)

2.2 pav. aktyvavimo funkcijos: a) sigmoidinė, b) slenkstinė, c) hiperbolinis tangentas, d) tiesinė

2.1.3 Vienasluoksnis neuroninis tinklas

Teoriškai vienas dirbtinis neuronas gali atlikti paprastus klasifikavimo uždavinius, bet sujungus neuronus į tinklą, klasifikacijos galimybės didėja. Paprasčiausias tinklas, kurį sudaro vienas sluoksnis neuronų, vadinamas perceptronu (2.7 pav.). Kairės pusės apskritimai yra įėjimo signalų skirstytuvai, nevykdantys jokių skaičiavimų. Kiekvienas įėjimo sluoksnio X elementas turi atitinkamą svorio koeficientą w_{jk} , kur jk nusako ryšį tarp j -tojo įėjimo ir k -tojo neuro.



2.3 pav. Vienasluoksnis neuroninis tinklas

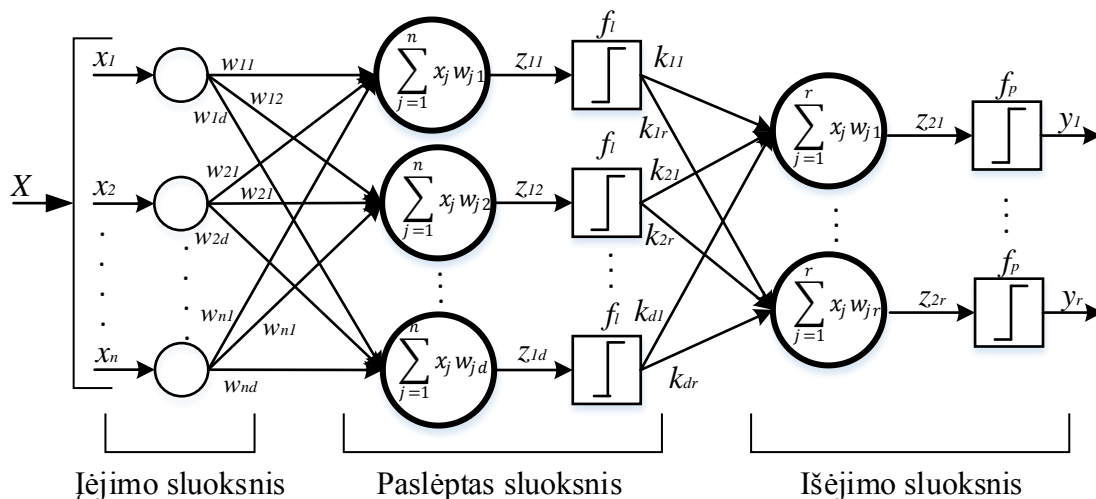
Kiekvienas išėjimas $y_j, j = 1, \dots, d$, yra įėjimo vektoriaus $X = (x_1, x_2, \dots, x_n)$ funkcija, kuri apskaičiuojama pagal formulę:

$$y_j = f(z_j) = f\left(\sum_{k=1}^n x_{jk} w_k\right), \quad (2.5)$$

čia d – neuroninio tinklo išėjimų skaičius, f – aktyvavimo funkcija (2.1) (2.2) (2.3) (2.4).

2.1.4 Daugiasluoksnis neuroninis tinklas

Neuroninis tinklas turintis daugiau nei vieną neuronų sluoksnį ir, kuriame nėra grįžtamojo ryšio, vadinamas daugiasluoksniu tiesioginio sklidimo neuroniniu tinklu (angl. *multilayer feedforward neural networks*) (2.8 pav.).



2.4 pav. Daugiasluoksnis neuroninis tinklas

2.8 paveiksle w_{ij} yra pirmojo neurono sluoksnio svorių koeficientai, k_{ij} yra antrojo neuronų sluoksnio svorių koeficientai. Taip pat skirtingiems neuronų sluoksniams galima taikyti skirtingas aktyvavimo funkcija.

Manykime, kad turime daugiasluoksnį neuroninį tinklą, kuriame yra L sluoksnių, $l = 0, 1, \dots, L$, kur $l = 0$ yra įėjimų sluoksnis o $l = L$ – išėjimų sluoksnis. Kiekvienas l sluoksnis turi n_l neuronų. Pirmojo sluoksnio išėjimo reikšmės y_j yra apskaičiuojamos pagal (2.5) formulę. l – ojo sluoksnio įėjimai į neuronus yra $(l - 1)$ – ojo sluoksnio išėjimai. Kiekvieno j – ojo neurono išėjimo reikšmė y_j l – jame sluoksnyje yra apskaičiuojama:

$$y_j = f(z_j) = f\left(\sum_{k=0}^{l-1} w_{jk} y_k\right), j = 1, \dots, n_l. \quad (2.6)$$

2.1.5 Neuroninio tinklo mokymas

Neuroninio tinklo mokomas tam, jog būtų minimizuoti neatitikimai tarp neuroninio tinklo ir realių rezultatų. Neuroninis tinklas gali būti mokomas trejopai:

- mokomas su norimomis išėjimo reikšmėmis;
- mokomas be išėjimo reikšmių;
- mišrus mokymas

Mokant su norimomis išėjimo reikšmėmis, įėjimo vektorius $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$ susiejamas su norimomis išėjimo reikšmėmis $T_i = (t_{i1}, t_{i2}, \dots, t_{id})$ $i = 1, \dots, h$. Tinklo išėjimo reikšmės, skaičiuojamos kiekvienam įėjimo vektoriui X_i , yra tiesiogiai susijusios su norimomis tų išėjimų reikšmėmis. Tinklas koreguojamas keičiant svorių vektorių reikšmes ir siekiant gauti kiek galima mažesnę paklaidą, t.y. ieškoma tokių svorių, kad skirtumas tarp norimų išėjimo reikšmių ir reikšmių gautų išmokius neuroninį tinklą būtų kiek galima mažesnės.

Kartais norimos gauti tinklo reikšmės nėra žinomos, tinklas mokomas be išėjimo reikšmių. Šio tipo metoduose tinklas mokomas ieškoti koreliacijų ar panašumų tarp mokymo aibės įėjimų. Čia nėra grįžtamojo ryšio, pasakančio, kuris atsakymas yra teisingas. Turima tik mokymo vektorių aibė X_i . Metodo tikslas suskirstyti mokymo duomenis į tam tikras klases arba rasti juose kokius nors reguliarumus arba ypatumus. Mišrus mokymas apima mokymą su norimais duomenimis ir be jų.

mokant vienasluoksnį tinklą svorių koeficientai keičiami taip, jog išėjimo vektorius $Y_i = (y_{i1}, y_{i2}, \dots, y_{in})$, gautas į įėjimą pateikus vektorių X_i , būtų kiek galima artimesnis vektoriui T_i , t.y. tinklo veikimo paklaida būtų kiek įmano mažesnė.

Paklaida $E(W)$ apibrėžiama kaip, svorių matricos $W = \{w_{ij}, i = 1, \dots, d, j = 0, \dots, n\}$, funkcija. Paklaidos funkcijos $E(W)$ minimumas randamas gradientiniais optimizavimo metodais.

Dažniausiai naudojamas gradientinis nusileidimo algoritmas, o paklaida apskaičiuojama taikant kvadratinių nuokrypių sumą:

$$E_i(W) = \frac{1}{2} \sum_{j=1}^d (y_{ij} - t_{ij})^2. \quad (2.7)$$

Mokant daugiasluksnį neuroninį tinklą reikia žinoti paslėptų neuronų išėjimų reikšmes. Jei paklaidos ir aktyvacijos funkcijos yra diferencijuojamos, ieškant minimalios paklaidos gali būti naudojama gradientinio nusileidimo metodas. Metodas, kuris naudoja gradientinį nusileidimą tinklo mokymui vadinamas „klaidos sklidimo atgal“ algoritmu (angl. *back-propagation learning algorithm*).

Šis algoritmas tinklą moko su norimomis išėjimų reikšmėmis, paklaida $E_i(W)$ apskaičiuojama pagal (2.7) formulę o svoriai keičiami pagal (2.8).

$$\Delta w_{jk}^i(t) = -\eta \frac{\partial E_i(t)}{\partial w_{jk}}, \quad (2.8)$$

čia η yra teigimas daugiklis (mokymo greitis), kuriuo reguliuojamas gradientinio optimizavimo žingsnio ilgis, t – iteracijos numeris. Pirmame algoritmo žingsnyje įėjimų vektoriui X_i apskaičiuojama išėjimų vektorius Y_i . Įvertinama paklaidos funkcija $E_i(W)$ išėjimų sluoksnyje L . Jei paklaidos funkcija $E_i(W)$ nelygi nuliui, jungčių svoriai keičiami pagal (2.8) formulę.

Dalinės išvestinės išreiškiamos formule:

$$\frac{\partial E_i}{\partial w_{jk}} = \frac{\partial E_i}{\partial z_{jk}} * \frac{\partial z_{jk}}{\partial w_{jk}} \quad (2.9)$$

Iš (2.6) gaunama

$$\frac{\partial z_{jk}}{\partial w_{jk}} = y_{ik}. \quad (2.10)$$

Tegul

$$\delta_{ij} = \frac{\partial E_i}{\partial z_{ij}} \quad (2.11)$$

Įstatę (2.11) ir (2.10) į (2.9) ir (2.8) formules gauname:

$$\Delta w_{jk}^i = -\eta \delta_{ij} y_{ik}, \quad (2.12)$$

čia j -asis neuronas priklauso l -ajam o k -asis neuronas priklauso $(l - 1)$ -ajam sluoksniui. Išėjimų sluoksnyje

$$\delta_{ij} = \frac{\partial E_i}{\partial z_{ij}} = f'(a_{ij})(y_{ij} - t_{ij}) \quad (2.13)$$

Kur j – asis neuronas priklauso L – ajam išėjimų sluoksniui.

Naudojant dalinės išvestinės $\frac{\partial E_i}{\partial w_{jk}}$ apibendrinę gauname :

$$\delta_{ij} = \frac{\partial E_i}{\partial w_{ij}} = \sum_{s=1}^{n_{l+1}} \frac{\partial E_i}{\partial z_{is}} * \frac{\partial z_{is}}{\partial z_{ij}} \quad (2.14)$$

čia n_{l+1} žymi neuronų (l+1)-ajame sluoksnyje skaičių. Išraiška $\frac{\partial E_i}{\partial z_{is}}$ yra lygi dydžiui δ_{ij} , apibrėžtam s-ajam neuronui (l+1)-ajame sluoksnyje. Atsižvelgiant į (2.6) gauname:

$$\frac{\partial z_{is}}{\partial z_{ij}} = f'(a_{ij})w_{is}. \quad (2.15)$$

Tada paslėptųjų j-ųjų neuronų

$$\delta_{ij} = f'(a_{ij}) \sum_{s=1}^{n_{l+1}} w_{is} \delta_{is} \quad (2.16)$$

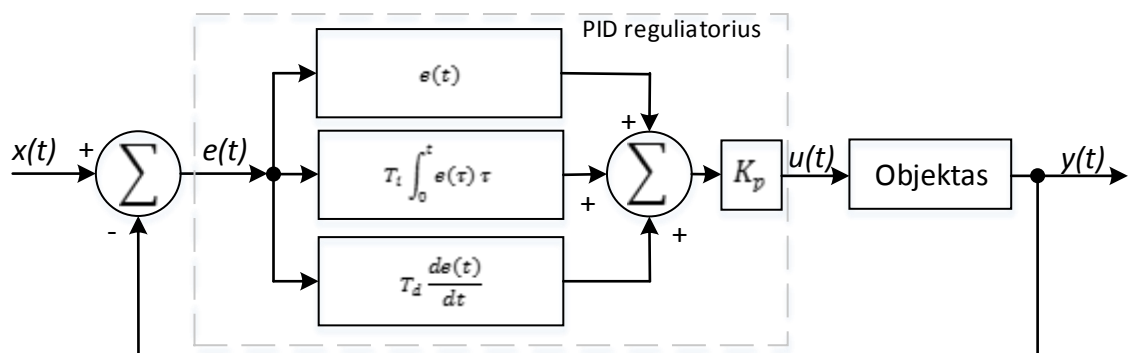
čia j-asis neuronas priklauso sluoksniui $l < L$, s-asis neuronas priklauso sluoksniui $l + 1$.

Iš pradžių apskaičiuojamas δ_{ij} reikšmės išėjimų sluoksnyje L pagal (2.14). Tada pagal (2.16) palaipsniui skaičiuojamos δ_{ij} reikšmės tarpiniuose sluoksniuose. Visi svoriai keičiami pagal (2.12). Tada į tinklą paduodamas naujas vektorius X_i ir mokymo algoritmas kartojamas iš naujo. Algoritmo sustojimo kriterijus yra arba iš anksto nustatyta paklaidos funkcijos slenksčio reikšmė, arba atitinkamas atliktų iteracijų (mokymo žingsnių) skaičius [12].

2.2 PID reguliatorius

Svarbiausias valdymo sistemos uždavinys – pasiekti ir palaikyti norimą sistemos būseną. Tai gali būti atliekama pasitelkus atviro kontūro sistemą, kurioje reguliatorius apdoroja užduoties signalą, arba uždaro kontūro sistemą, kurioje reguliatorius apdoroja užduoties ir objekto išėjimo signalų skirtumą. Uždaro kontūro sistemos užduotis palaikyti kuo mažesnę užduoties ir įėjimo signalų skirtumą, nepaisant objekto dinaminė savybių.

Standartinė uždaro kontūro sistema pavaizduota 2.5 paveiksle. Sistemos valdymo signalas – $x(t)$, $y(t)$ yra objekto išėjimo signalas, $e(t)$ valdymo paklaida, kuri lygi $e(t) = x(t) - y(t)$. Objekto valdymo signalas – $u(t)$.



2.5 pav. Uždaro kontūro valdymo sistema su PID reguliatoriumi

Labiausiai paplitusi lygiagrečioji reguliatoriaus struktūra, kur $u(t)$ reiškiamas:

$$u(t) = K_p \left[e(t) + T_i \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right], \quad (2.17)$$

čia K_p yra proporcinės grandies stiprinimo koeficientas, T_i – integruojančios grandies laiko pastovioji, T_d – diferencijuojančios grandies laiko pastovioji.

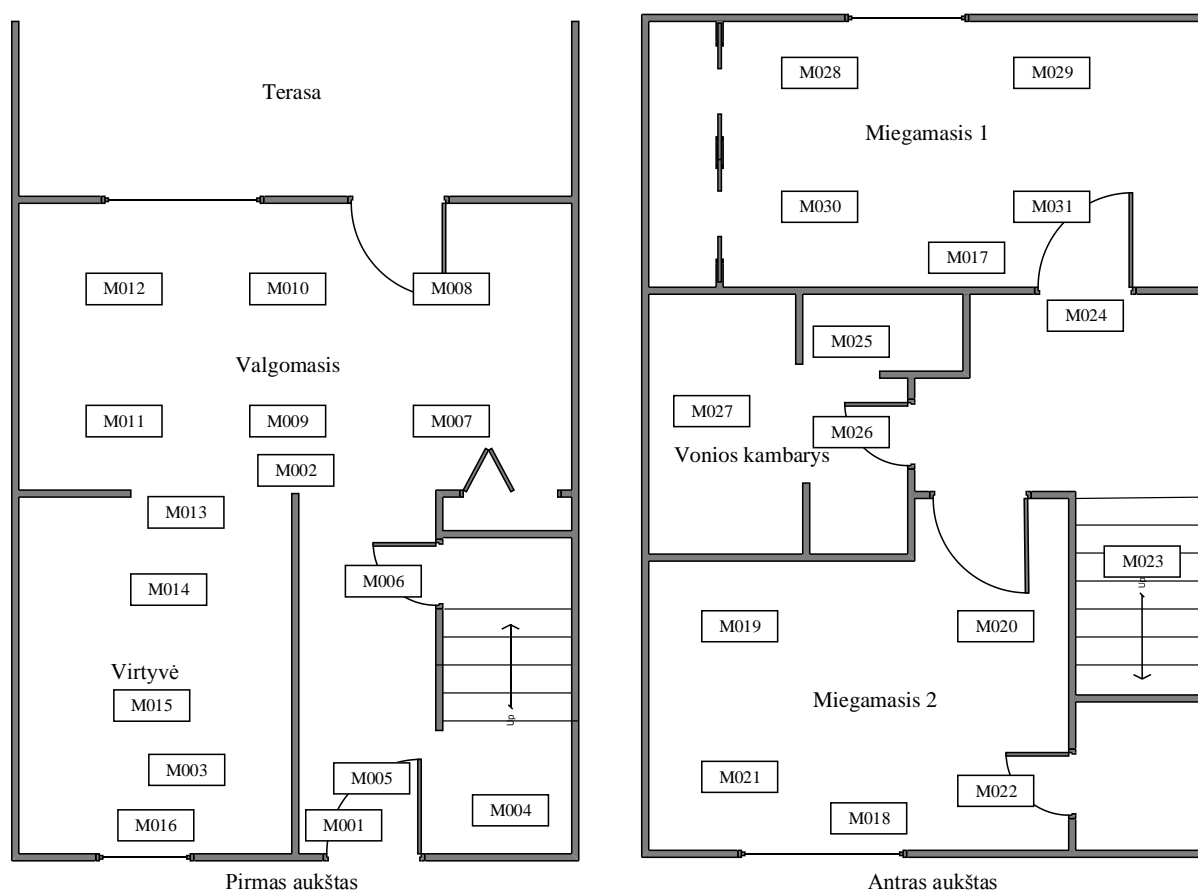
Lygtį sudaro proporcinė, integralinė ir diferencialinė dedamosios. Reguliatorių gali sudaryti įvairūs dedamųjų variantai, kurie priklauso norimų objekto dinaminių savybių ir reguliavimo charakteristikų.

Proporcinio regulatoriaus valdymo signalas proporcingas sistemos nuokrypiui. Jis tiesiogiai priklauso nuo sistemos nuokrypio ir jeigu $e(t)$ lygu 0, nėra formuojamas valdymo signalas $u(t)$. Tai lemia, kad dėl proporcinio regulatoriaus atsiranda statinė sistemos paklaida. Integruojantis reguliatorius integruoja sistemos paklaidą laiko intervale $(0; t)$. Esant pastoviam sistemos nuokrypiui, laikui bėgant nuokrypio suma didėja ir integruojantis reguliatorius keičia valdymo signalą taip, jog nusistovėjęs nuokrypis ir nuokrypio suma mažėtų. Diferencijuojanti grandis nusako, koku greičiu kinta sistemos nuokrypis. Ši grandis sumažina sistemos perreguliavimą, tačiau kaip atskiras reguliatorius nėra naudojamas [13].

3. IŠMANIOJO TERMOSTATO PROGRAMINĖ REALIZACIJA

3.1 Sistemos mokymui skirtų duomenų apdorojimas

Sistemai realizuoti, buvo panaudoti Tulum išmanaus būsto duomenys (angl. *Tulum smart apartment*) (TSA) [14]. Duomenys buvo kaupiami apie kasdieninį sutuoktinių gyvenimą, 2009 – 2010 akademisiais metais. Dviejų aukštų pastate (2 pav.) sumontuota trisdešimt vienas judesio jutiklis M001 – M031 ir penki temperatūros jutikliai T001 – T005.



3.1 pav. Tulum pasato pirmo ir antro aukšto planas su judesio jutiklių išdėstymu.

3.1 lentelėje pavaizduota TSA duomenų fragmentas, juos sudaro: einamosios dienos data, einamasis laikas, jutiklių identifikacijos numeriai, jutiklių reikšmės ir gyventojų veikla.

3.1 lentelė. TSA duomenų fragmentas

| Data | Laikas | Jutiklis | Reikšmė | Veikla |
|------------|-----------------|----------|---------|--------------------------|
| 2009-09-25 | 13:05:04.022174 | T004 | 24.5 | |
| 2009-09-25 | 13:06:01.084545 | M019 | ON | R1_Sleeping_in_Bed begin |
| 2009-09-25 | 13:06:06.043407 | M019 | OFF | |
| 2009-09-25 | 13:06:07.037925 | M019 | ON | |

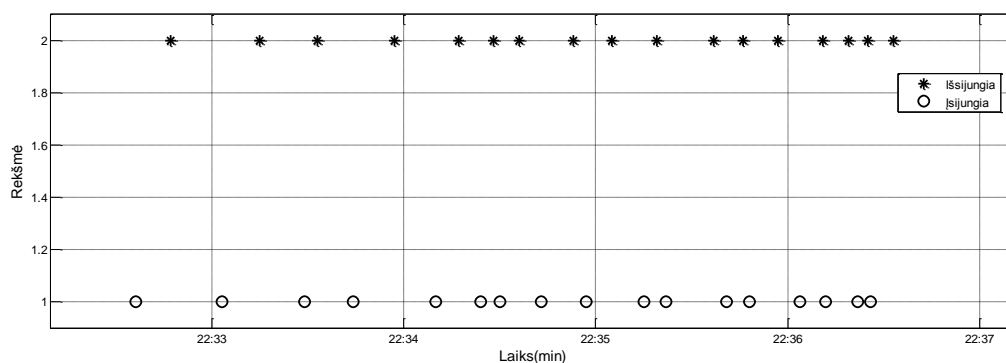
Iš turimų TSA duomenų suformuojamos keturios matricos:

- laiko matrica – *Laikas*;
- jutiklių reikšmių matrica – *M*;
- gyventojų miego matrica – *Miegas*;
- gyventojų išvykimo matrica – *Isvyk*.

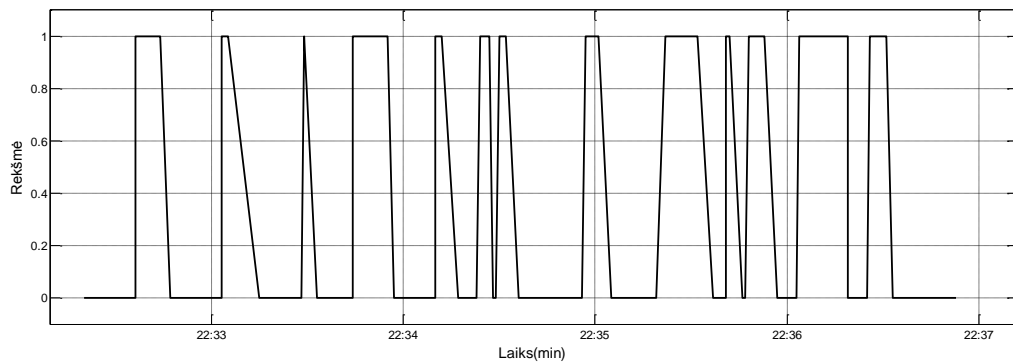
TSA duomenys padalinami į datas, laiko, jutiklių, reikšmių ir veiklos matricas. Laiko matricai suformuoti panaudojamos „strcat“ ir „datetime“ Matlab funkcijos. „strcat“ funkcija sujungia datos ir laiko reikšmes o „datetime“ sujungtą datą ir laiką paverčia į skaičių.

M matrica suformuojama pasinaudojus algoritmu, pavaizduotu 3.2 paveiksle. *M* matricą sudaro 31 stulpelis, kiekvienas matricos stulpelis, pagal savo numerį, atitinka judesio jutiklio identifikacijos numerį ir jo reikšmes. Pavyzdžiui, pirmas *M* matricos stulpelis atitinka M001 judesio jutiklio duomenis. Pirmiausiai pasinaudojus „strcmp“ funkcija, suformuojamos loginės *true* ir *false* matricos, kurios sudarytos iš 1 ir 0. *true* matricos reikšmę 1 atitinka reikšmių matricos „ON“ momentus, o *false* matricoje 1 atitinka reikšmių matricos „OFF“ momentus. Toliau pasinaudojus ta pačia „strcmp“ funkcija sukuriama *x* matrica, kuri atitinka jutiklių matricos *jutikliai_j* reikšmes, čia $jutikliai = \{M001, \dots, M031\}$. *true* ir *false* matricos, naudojantis logine sandauga \wedge , padauginamos iš *x* matricos. Loginė $true \wedge x$ sandauga nurodo, kur įsijungia konkretus judesio jutiklis, o $false \wedge x$ nurodo jutiklio išsijungimą. *A* matricą sudaro konkretaus jutiklio įsijungimo ir išsijungimo momentai, kurie atitinkamai žymimi 1 ir 2 (3 pav.).

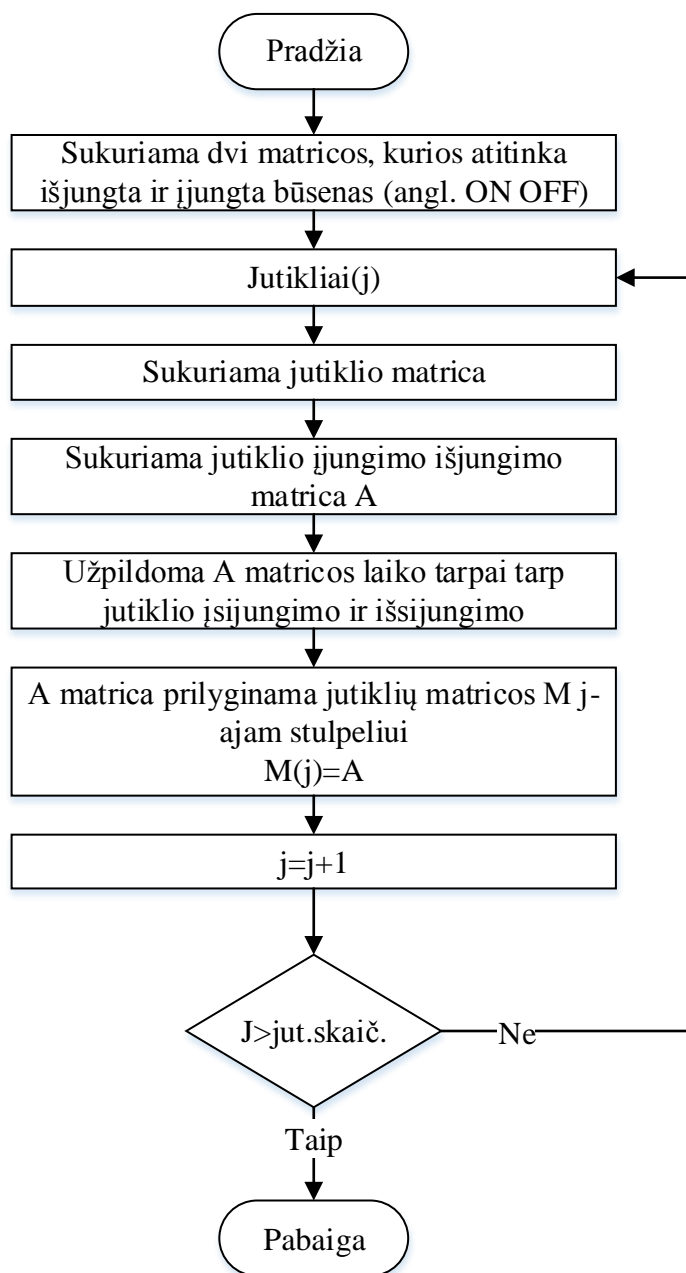
Tam, jog *A* matrica vaizduotų ne įsijungimo ir išsijungimo momentus, o jutiklių veikimo ir neveikimo intervalus (3.3 pav.), panaudojamas 3.5 paveiksle pavaizduotas algoritmas. *Matlab* išeities kodas *M* matricai sudaryti pateiktas 1 priede.



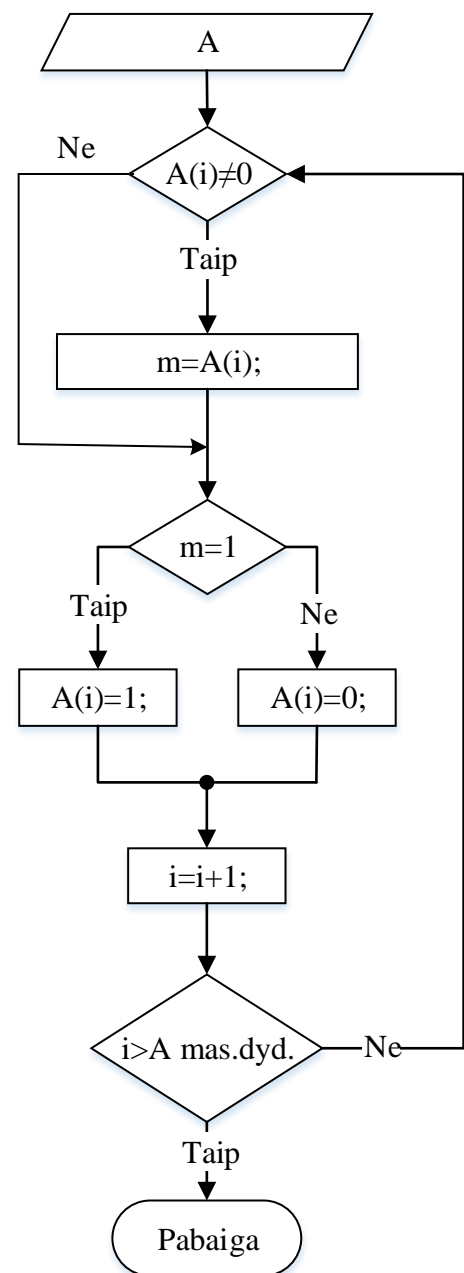
3.2 pav. M 15 jutiklio 5 minučių reikšmės



3.3 pav. M015 jutiklio duomenys apdoroti laiko tarpų užpildymo algoritmo

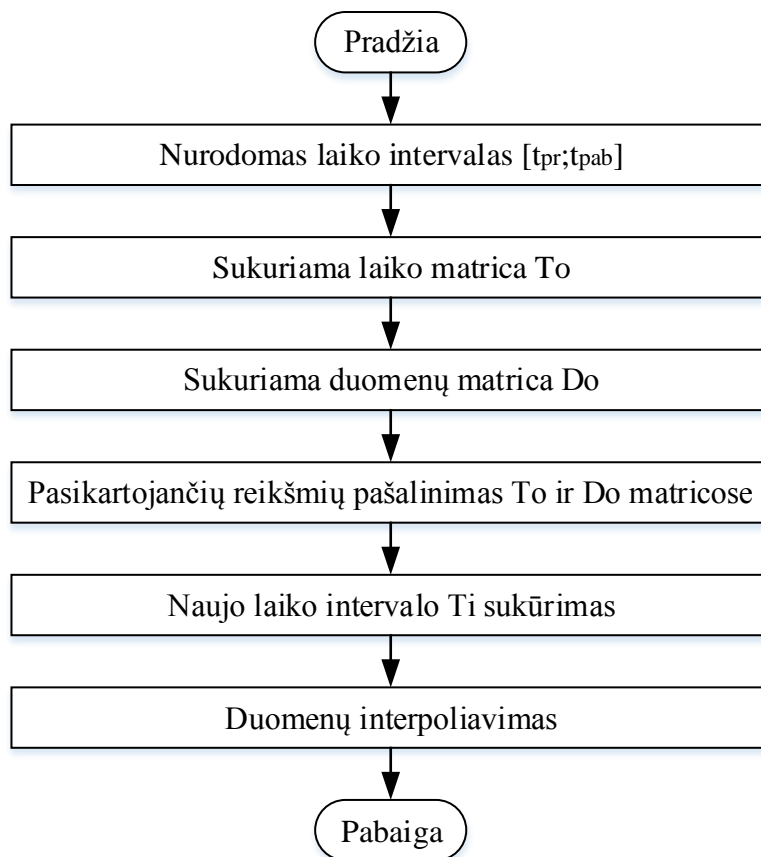


3.4 pav. M matricos formavimo algoritmas



3.5 pav. Laiko tarpų užpildymo algoritmas

TSA duomenyse yra fiksuojama jutiklių įsijungimo ir išsijungimų momentai. Kaip pateikta 1 lentelėje, laiko intervalai yra atsitiktiniai. Tam, kad nuskaitymo reikšmės būtų periodinės, reikia interpoliuoti duomenis. 3.6 paveiksle pateikta duomenų interpoliavimo algoritmas. Jo išeitie *Matlab* kodas pateiktas 2 priede.



3.6 pav. Duomenų interpoliavimo algoritmas

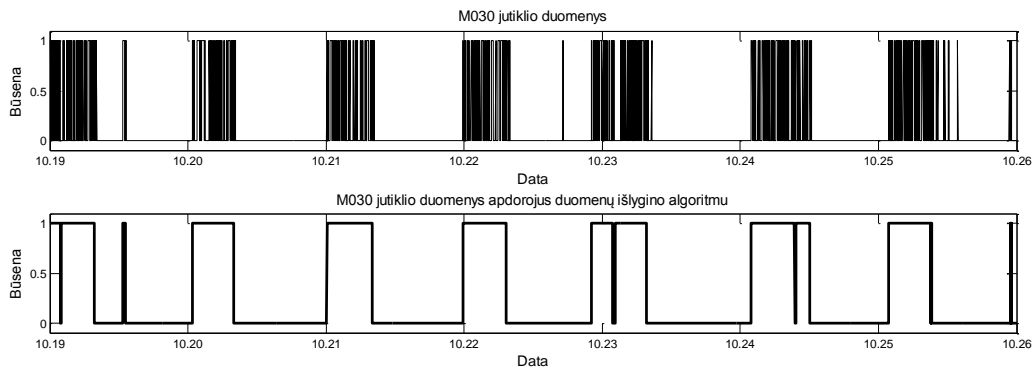
TSA duomenys yra sugrupuojami savaitėmis. Atmetus savaites, kuriose trūksta bent vienos dienos duomenų, gaunama šešiolika pilnų savaitių. Interpoliuojant duomenis pirmiausiai nurodoma savaitės, kurios duomenis norima suinterpoliuoti, pradžia t_{pr} ir pabaiga t_{pb} . Toliau sukuriamos To ir Do matricos:

- $To = Laikas(t_{pr}; t_{pb});$
- $Do = M(t_{pr}; t_{pb}).$

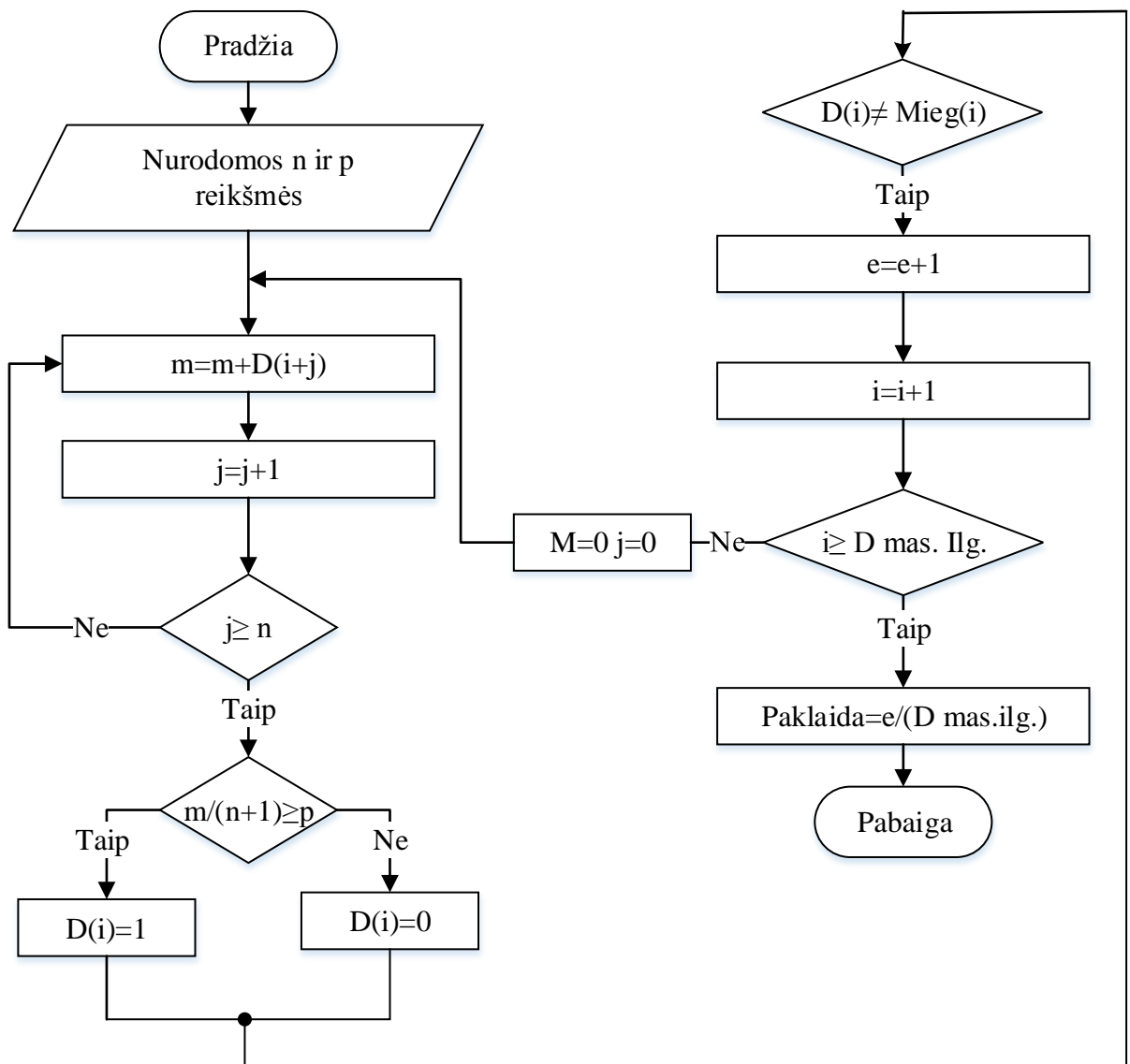
Tam, jog nepriklausomai nuo pasirinktos savaitės, suinterpoliuotų duomenų atskaitų skaičius būtų vienodas, savaitės prasidėtų ir baigtųsi vienu laiku, prie To matricos pradžios ir pabaigos pridedamas 00:00:00 laikas. Do matricos pradžia ir pabaiga papildoma nulio reikšmėmis.

Interpoliuojant duomenis su Matlab funkcija „inperp1“ negali pasikartoti duomenys, t.y. per sekunde galimas tik vienas nuskaitymas. Pašalinti pasikartojančius duomenis naudojama „unique“ Matlab funkcija. Interpoliuojant duomenis be To ir Do dar nurodoma nauja laiko matrica Ti su pastoviu vienos minutės laiko intervalu.

Jutiklių veikimo intervalai yra trumpi (3.3 pav.). Tam, kad iš jutiklių eitų iširti konkretų pastato užimtumo tipą, reikia jog jutiklių reikšmės būtų kiek įmano tolydesnės. Panaudojamas 3.8 paveiksle pavaizduotą jutiklių duomenų išlygino algoritmą jutiklių reikšmės yra išlyginamos (3.7 pav.). *Matlab* išeities kodas pateiktas 3 priede.



3.7 pav. M030 jutiklio originalūs ir išlyginti duomenys



3.8 pav. Jutiklių duomenų išlygino algoritmas

Algoritmas kiekvieną $D(t_i)$ reikšmę prilygina 1 arba 0, priklausomai nuo laiko intervale $[t_i; t_{i+n}]$ esančių teigiamų D matricos reikšmių. Šio algoritmo tikslumą lemia n ir p kintamieji – n kintamasis nurodo kiek minučių į prie bus skaičiuojamos teigiamos D matricos reikšmės, o p kintamasis nurodo, kokiam teigiamų D matricos reikšmių santykiui su bendru laiko intervalo $[t_i; t_{i+n}]$ reikšmių skaičiumi esant, $D(t_i)$ yra lygu 1, priešingu atveju $D(t_i)$ lygu 0.

3.2 Pastato užimtumo timpų išskyrimas iš TSA duomenų

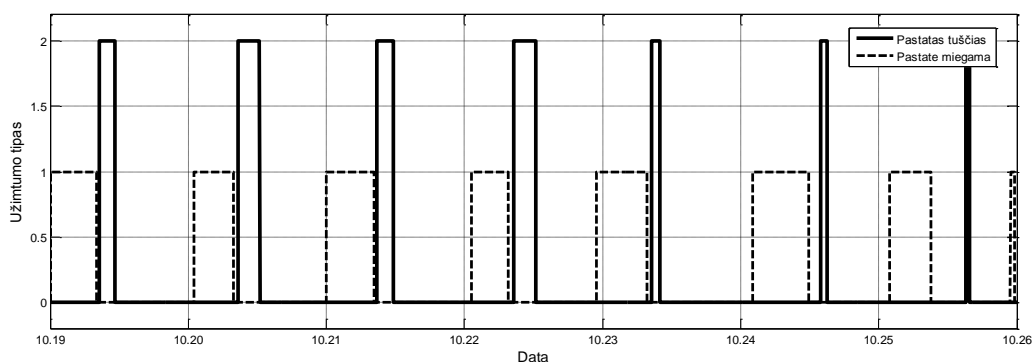
Norint pasiekti mažiausias energijos sąnaudas šildymui, reikia išskirti tris pastato užimtumo tipus:

- gyventojai miega;
- gyventojai išvykę;
- gyventojai namuose.

Pasinaudojant veiklos matrica galima nustatyti du iš trijų pastato užimtumo tipus, trečias tipas (gyventojai namuose) bus tada, kai gyventojai nei miegos, nei bus išvykę. Išvykimo pradžia bus laikoma „*Leave_Home end*“, kai gyventojai baigia ruoštis ir išvyksta iš namų. Išvykimo pabaiga bus laikoma „*Enter_Home begin*“, kai gyventojai įžengia į namus. Pasinaudojant 3.4 ir 3.5 paveiksluose pateiktais algoritmais sukuriama *Išvyk* matrica, kuri nurodo, laiko intervalus kuriais gyventojai yra išvykę.

Norint nustatyti, kada pastate yra miegama, reikia atkreipti dėmesį, jog TSA duomenyse nurodoma pirmo ir antro asmens miego pradžia ir pabaiga atskirai. Pirmo asmens miego pradžia fiksuojama „*R1_Sleeping_in_Bed begin*“, miego pabaiga – „*R1_Sleeping_in_Bed end*“. Antrojo asmens miego pradžia – „*R2_Sleeping_in_Bed begin*“, pabaiga – „*R2_Sleeping_in_Bed end*“. Kadangi asmenų miegojimo laikas nesutampa, Matricą *Miegas* sudarys laiko intervalai, kai abu asmenys miega t.y. persidengiantys laiko intervalai.

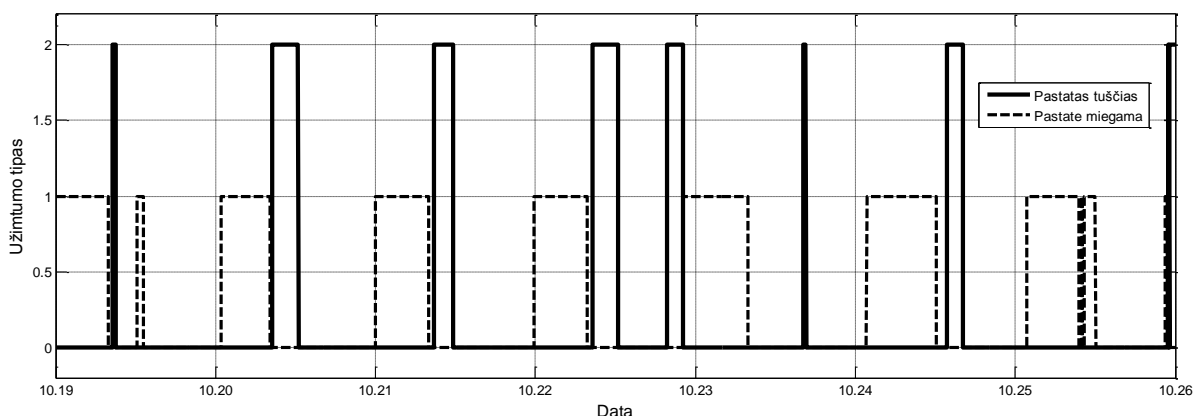
Suformavus *Išvyk* ir *Miegas* matricas, galima išskirti tris pastato užimtumo tipus (3.9 pav.). Ištisinė linija žymi kada gyventojai išvykę, punktyrinė – kada miega. Nulinės grafiko reikšmės reiškia, jog gyventojai yra namuose.



3.9pav. Pastato užimtumas

Kitas etapas – rasti judesio jutiklius, kurių veikimo ir neveikimo intervalai atitinka konkrečias gyventojų veiklas. Tokiu būdu, turint pastate sumontuotų judesio jutiklių duomenis, būtų galima išskirti tam tikrus pastato užimtumo tipus. Žinant pastato planą ir judesio jutiklių išdėstymą (3.1 pav.), išskiriami jutikliai, kurių reikšmės atitinka konkrečias veiklas.

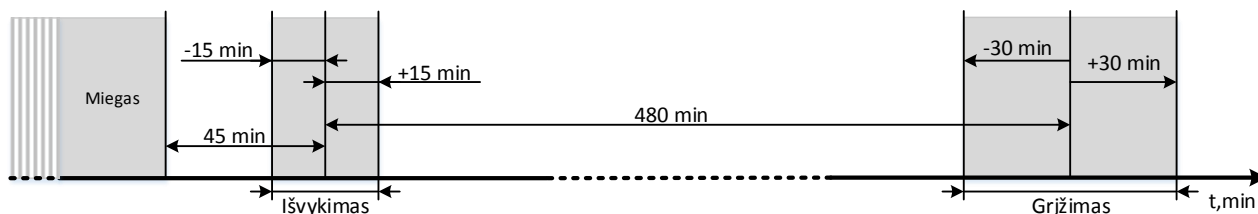
Tam, kad tiksliau būtų galima nustatyti pastato užimtumo tipus, jungiama kelių jutiklių duomenys ir lyginami su atitinkamu užimtumu gautų iš gyventojų veiklos. Nustatyti, jog gyventojai miega, naudojami įvairūs miegamojo jutiklių jungimo variantai, o norint nustatyti išvykimą laikoma, kad tai bus laiko intervalai, kai pastate nebus fiksuojami judesiai. 3.10 paveiksle vaizduojami pastato užimtumo tipai, išskirti iš judesio jutiklių duomenų.



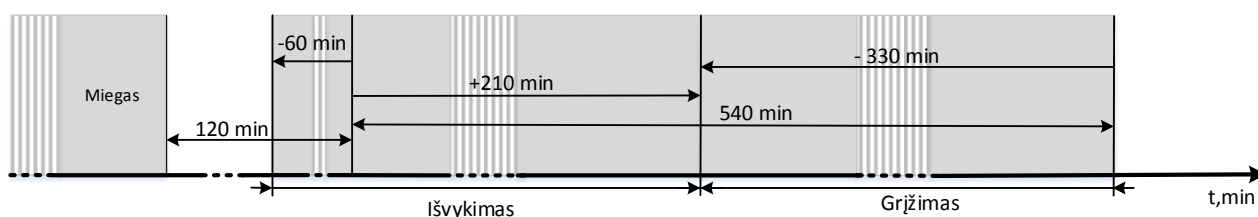
3.10 pav. Pastato užimtumo tipai, išskirti iš judesio jutiklių duomenų

Gyventojų miegas yra gana pastovus: vidutiniškai trunka 8 h ir 30 min. Tačiau gyventojai iš pastato išvyksta nesistemiškai ir trumpais laiko intervalais.

Norint sistemą išbandyti, kai gyventojai išvyksta sistemingai, sukuriama duomenų matrica, kuri atitiktų, nuosekliu gyventojų išvykimo intervalus. Šie laiko intervalai modeliuojami darbo dienoms (3.11 pav.) ir savaitgaliams (3.12 pav.) atskirai.



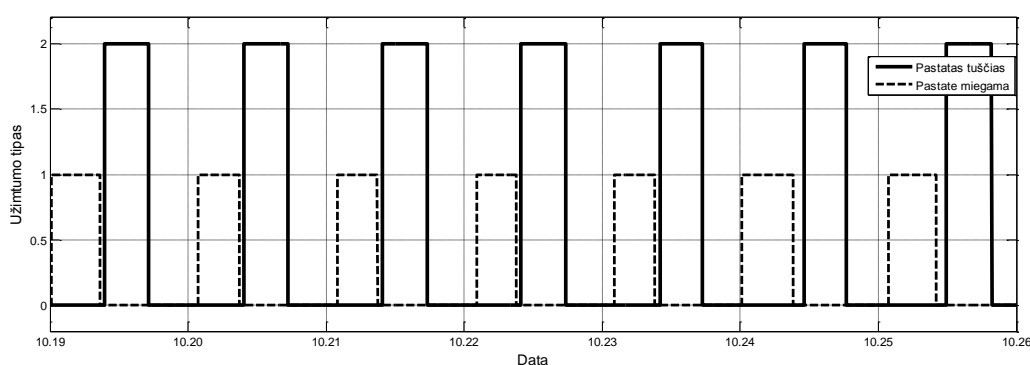
3.11 pav. Darbo dienų išvykimo modelis



3.12 pav. Savaitgalių išvykimo modelis

Priimama, kad išvykimo intervalai priklauso nuo gyventojų miego, t.y. gyventojai išvyksta iš namų praėjus tam tikram laiko tarpui nuo atsikėlimo. Modeliuojama, kad normaliai darbo dienomis gyventojai išvyksta 8 h. Išvykimo pradžia prasideda 45 min po atsikėlimo. Norint išvykimą ir grįžimą namo padaryti dinamiškesnį, išvykimui ir grįžimui nustatomi galimi kitimo režiai – išvykimo $[-15; 15]$, grįžimo $[-30; 30]$.

Analogiškai sukuriama savaitgalio išvykimo intervalai. Nustatoma 9 valandų maksimalus išvykimas, tačiau numatant, jog savaitgaliais gyventojų veikla mažiau sisteminga, todėl kitimo režiai parenkami taip, jog gyventojai savaitgaliais gali ir neišvykti: išvykimo $[-60; 210]$, grįžimo $[-330; 0]$ kitimo režiai. Sumodeliuoti išvykimai ir gyventojų miego intervalai pateikti 3.13 paveiksle.



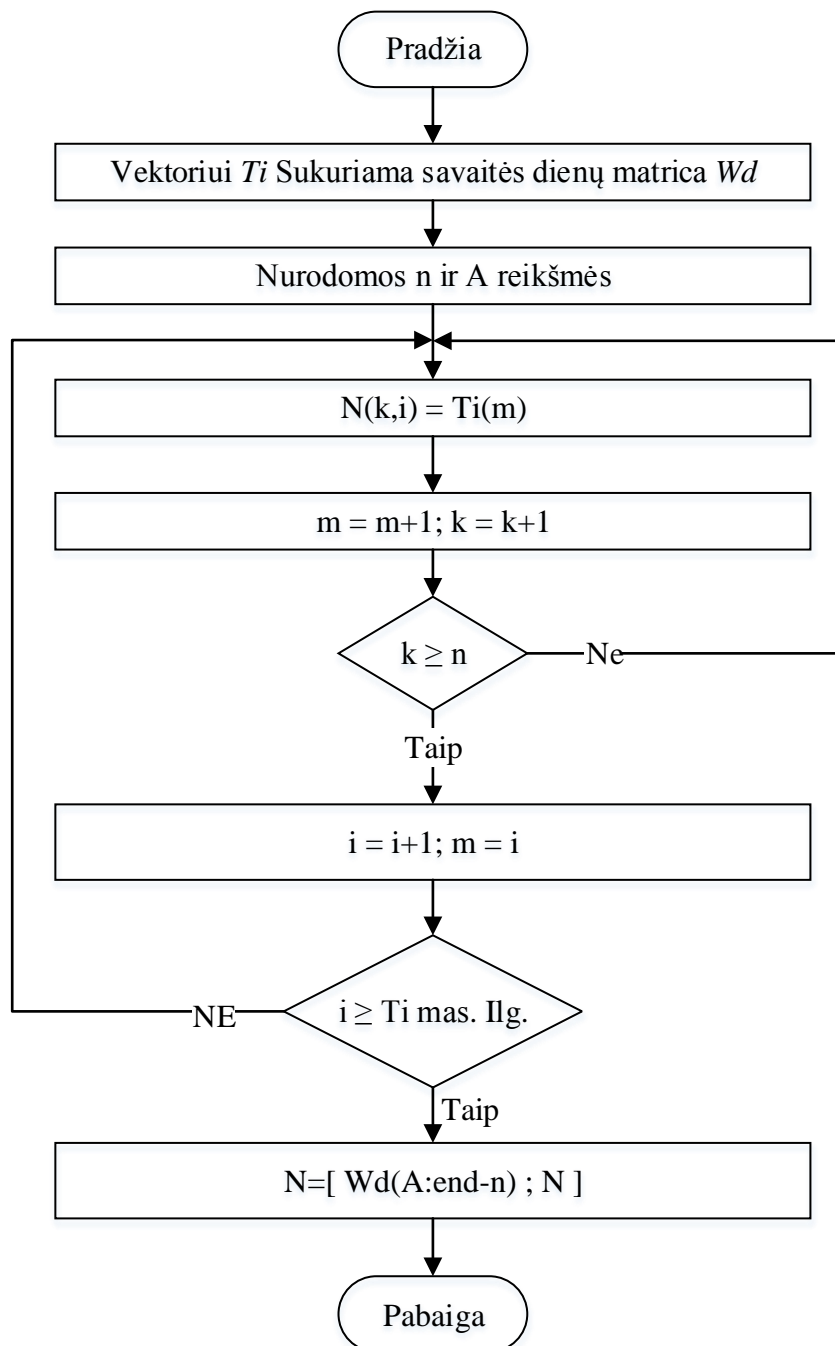
3.13 pav. Pastato užimtumo tipai, esant sumodeliuotam išvykimui

3.3 Neuroninio tinklo mokymas

Neuroninio tinklo mokymui reikia sudaryti įėjimų N ir norimų išėjimų T matricas. 3.14 paveiksle pavaizduotas įėjimų sudarymo algoritmas, kurio *Matlab* išeities kodas pateiktas 4 priede. Įėjimų matricą sudaro n eilučių ir m stulpelių. Stulpelį St_i sudaro n laiko reikšmių: $St_i = t_i, t_{i+1}, \dots, t_{i+n}$.

Norimų išėjimų T matricą sudaro trys stulpeliai. Kiekvieno stulpelio teigiamas elementas „1“, atitinka vieną iš trijų pastato užimtumo tipų. Pirmo T matricos stulpelio reikšmės atitinka A minučių į priekį perslinktus pastato užimtumo tipų reikšmes

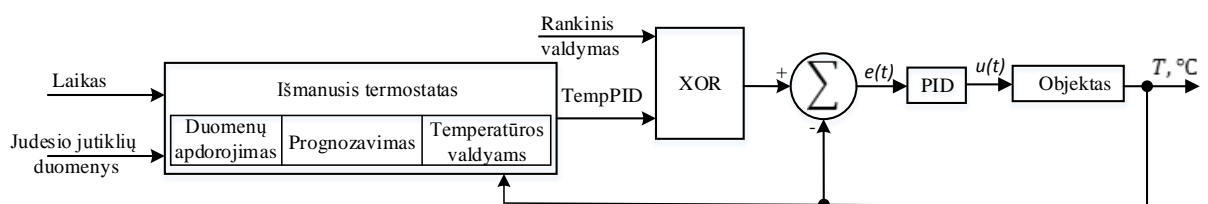
$$T = \begin{bmatrix} Namie(b_{i+A}, \dots, b_z) \\ Miega(b_{i+A}, \dots, b_z) \\ Išvykę(b_{i+A}, \dots, b_z) \end{bmatrix}.$$



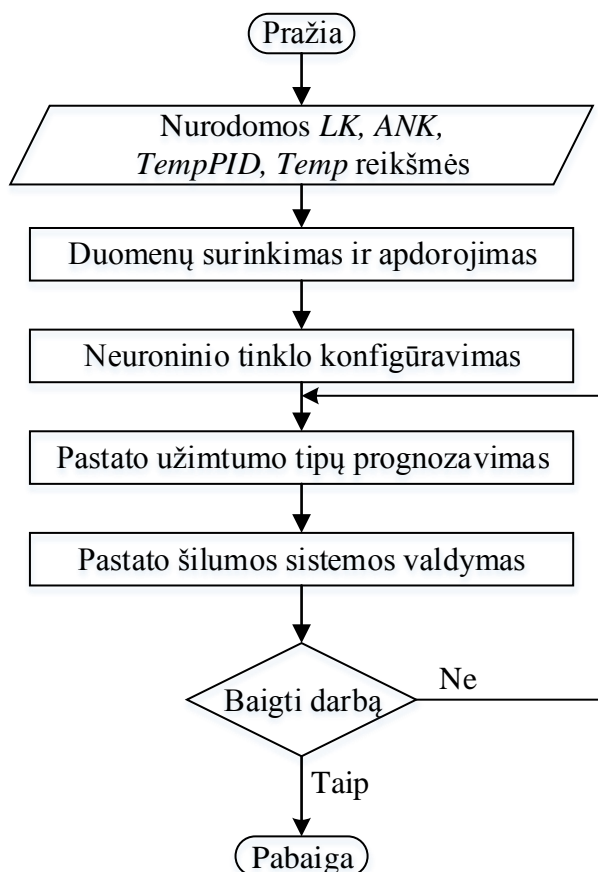
3.14 pav. neuroninio tinklo įėjimo sudarymo algoritmas.

3.4 Išmaniojo termostato veikimas

Pastato šildymo sistema su išmaniuoju termostatu ir jo veikimo algoritmas pateikti atitinkamai 3.15 ir 3.16 paveiksluose, visas išeities kodas pateiktas 5 priede.



3.15 pav. Pastato šildymo funkcinė blokinė schema su išmaniuoju termostatu



3.16 pav. Išmaniojo termostato veikimo algoritmas

Sistemos veikia penkiais etapais, iš kurių trys yra atliekami tik įjungus sistemą. Pirmiausiai sistemai nurodoma pradiniai duomenys:

- laiką LK , kurio reikia pakelti pastato temperatūrą 1°C ,
- galimą išankstinį šildymo sistemos išjungimo laiką ANK ,
- pradinę temperatūrą $TempPID$, kurią palaikys termostatas, kol bus nustatytas pastato prognozuojamas užimtumo tipas,
- temperatūras $Temp$, būdingas atitinkamam pastato užimtumo tipui.

Sukauptus judesio jutiklių duomenis sistema apdoroja ir išskiria pastato užimtumo tipus (plačiau aprašyta 3.1 skyriuje) ir su turima pastato užimtumo informacija mokomas neuroninis tinklas.

Tinklo mokymui naudojamos dvi Matlab funkcijos:

```

net = feedforwardnet(cc),
net = train(net,N,T);

```

„*feedforwardnet*“ funkcija parenkamas tiesioginio sklidimo tinklo tipas, cc yra vektorius nurodantis paslėptų sluoksnių ir neuronų skaičių. Funkcija „*train*“ mokomas tinklas taip, jog esant N įėjimo vektoriui norimos tinklo išėjimo reikšmės y būtų kuo artimesnės matricos T reikšmėms. N ir T matricių sudarymas plačiau aptartas 3.2 skyriuje.

Sistema, pastato užimtumas, prognozuoja A minučių į priekį. Fiksuodama užimtumo tipą laiko momentais $(t + A - 1)$ ir $(t + A)$, čia t – einamasis laikas, tikrina, ar nenumatomas užimtumo tipo pokytis. Kiekvieną užimtumą termostatas atpažįsta kaip atitinkama kintamojo c reikšmę:

- 1 – pastatas užimtas;
- 2 – pastate miegama;
- 3 – pastatas tuščias.

Išmanojo termostato *Matlab* kodo fragmentas kuris išskiria pastato užimtumo tipą:

```
for j=1:3
if y(j,1)>q
q=y(j,1);
c=j;
end
end
```

Neuroninio tinklo išėjimo vektorių y sudaro trys tikimybės: p_1, p_2, p_3 , kurios nusako kiekvieno pastato užimtumo tikėtinumą, esant N įėjimo vektoriui. Parenkamas tas užimtumo tipas, kurio tikimybė didžiausia.

Užfiksavus užimtumo pasikeitimą, tikrinamas pokyčio tipas: kylantis ($a=1$) ar krintantis frontas ($a=2$). Krintantis frontas fiksuojamas, kai:

- $b(t+A-1)=2$ & $b(t+A)=1$;
- $b(t+A-1)=3$ & $b(t+A)=1$;
- $b(t+A-1)=3$ & $b(t+A)=2$.

Kylantis frontas fiksuojamas, kai:

- $b(t+A-1)=1$ & $b(t+A) =2$;
- $b(t+A-1)=1$ & $b(t+A) =3$;
- $b(t+A-1)=2$ & $b(t+A) =3$.

Termostatas prognozuoja A minučių užimtumo tipų pokyčius ir, jeigu fiksuojamas užimtumo tipo pasikeitimas, į matricą *info* surašo šiuos duomenis: pasikeitimo prioritetą, pasikeitusiam užimtumui būdingą temperatūrą *TempX* ir pokyčio laiko momentas *Lx*.

Išmanojo termostato *Matlab* kodo fragmentas kuris formuoja *info* matricą:

```
if a~=0
info(1,A)=a;
info(2,A)=Temp(c,1);
info(3,A)=t+1/12;
else
info(:,A)=[0;0;0];
end
```

Kiekviename cikle sistema fiksuoja artimiausią užimtumo pokytį ir apskaičiuoja:

- $TempX$ ir esamos pastato temperatūros $YTemp$ skirtumą $dTemp$;
- laiką Ly , reikalingą pastato temperatūrai pakelti iki $TempX$;
- Laiką dL , kuris likęs iki užimtumo pasikeitimo.

Išmaniojo termostato *Matlab* kodo fragmentas, kuris fiksuoja info matricos duomenis:

```
[row col] = find(info);  
Prio=info(row(1,1),col(1,1));  
TempX=info(row(2,1),col(1,1));  
Lx=info(row(3,1),col(1,1));  
dTemp = TempX-YTemp(end,1);  
Ly = LK*dTemp;  
Li=t;  
dL=Lx-t;
```

Matlab funkcija „*find*“ suranda *info* matricos reikšmes, nelygias nuliui, ir nurodo jų vietas matricoje. Kadangi sistema fiksuoja artimiausią užimtumo pasikeitimą, tai naudojama pirma „*find*“ funkcijos surasta reikšmė.

Išmaniojo termostato temperatūros parinkimo *Matlab* kodo fragmentas:

```
if Prio==1 && Ly >=dL  
TempPID = TempX;  
else if Prio==2 && Li>=(Lx-ANK )  
TempPID = TempX;  
end  
end
```

Priklausomai nuo užimtumo tipo pokyčio prioriteto *Prio* termostatas parenka laiko momentą t , kai bus keičiama temperatūra $TempPID$. Termostato prioritetas yra užtikrinti malonią temperatūrą gyventojams, todėl pirmumas teikiamas tokiu atveju, kai užimtumo tipui būdinga aukštesnė temperatūra, t.y. kai fiksuojamas krintantis frontas. Jeigu numatomas aukštas užimtumo pokyčio prioritetas, sistema tikrina ar laikas Ly nėra didesnis ar lygus dL . Termostatas keičia temperatūrą, jeigu laikas, kuris liko iki užimtumo pasikeitimo, lygus ar mažesnis už Ly .

Jeigu numatomo pokyčio prioritetas žemas (kylantis frontas), termostatas išjungs šildymo sistemą, kai $dL \geq (Lx - ANK)$, čia ANK – išankstinio išjungimo laikas minutėmis.

4. IŠMANIOJO TERMOSTATO VEIKIMO EKSPERIMENTINIAI TYRIMAI

Ekspimentinių tyrimų metu buvo fiksuojamas išmaniojo termostato veikimo tikslumas. Termostatas veikia trimis etapais: duomenų apdorojimo, prognozavimo ir valdymo. Pirmų dviejų etapų atlikimo tikslumas lemia, kaip tiksliai sistema nustatys pastato šilumą reikiamu laiku. Tyrimams buvo naudojama *Matlab* 8.3 programinė įranga ir personalinis kompiuteris (AMD A8 –4500M 2,8Ghz, 8G RAM).

4.1 Duomenų apdorojimo tikslumas

Sistema buvo mokoma trijų tipų duomenimis:

1. duomenimis suformuotais iš gyventojų veiklos;
2. duomenimis suformuotais iš judesio jutiklių reikšmių;
3. sumodeliuotais duomenimis su sistemingais užimtumo tipais.

Buvo laikomasi nuomonės, jog duomenys suformuoti iš gyventojų veiklos yra tikslūs ir visiškai atitinka namo užimtumo tipus. Jais remiantis buvo fiksuojamas duomenų, suformuotų iš judesio jutiklių, tikslumas.

Nustatant pastato užimtumo tipą gyventojams miegant, pastebėta, kad pirmojo miegamojo judesio jutiklių (3.1 pav.) veikimo reikšmės atitinka gyventojų miegą. Tiriant, kaip judesio jutiklių reikšmės atitinka minėtam užimtumo tipą buvo jungiamos įvairių jutiklių reikšmės ir tikrinama, kaip jos atitinka gyventojų miegą (4.1 lentelė).

4.1 lentelė. Paklaidos priklausomybė nuo jungiamų judesio jutiklių reikšmių.

| Jutiklių nr. | 17,28,29,30,31 | 17,29,30,31 | 17,30,31 | 30,31 | 30 | 31 |
|--------------------|----------------|-------------|----------|-------|--------|--------|
| <i>n</i> | 69 | 70 | 69 | 68 | 78 | 73 |
| <i>p</i> | 0,15 | 0,15 | 0,10 | 0,10 | 0,1 | 0,05 |
| Paklaida | 10,88% | 11,32% | 11,03% | 12,9% | 12,17% | 28,71% |
| Paklaida min/dieną | 156 | 163 | 159 | 185 | 209 | 395 |

Paklaida buvo gauta mažiausia (10,88%), kai buvo sujungiamos visos miegamojo jutiklių reikšmės, *n* parinkta 69 o *p* – 0,15. Vidutinis vienos dienos netiksliai identifikuotos miego veiklos laikas yra 156 minutės. 86 minutes nustatoma veikla, kada jos nėra, ir 72 minutes nenustatoma, tačiau yra miegama. Remiantis TSA duomenyse pateikta gyventojų veikla, vidutiniškai miegama 8h 30min, naudojant pirmo miegamojo jutiklių duomenis, tiksliai atpažįstama 7h (82,35%) gyventojų miego.

Norint nustatyti išvykimą manoma, jog tai bus laiko intervalai, kai pastate nebus fiksuojami judesiai. Remiantis veiklos duomenimis vidutiniškai gyventojai būna išvykę iš namų

dvi valandas. Sujungus visus pastato judesio jutiklius ir apdorojant duomenis išvykimas nustatomas 91,4% tikslumu, tačiau žinant, jog gyventojai būna išvykę tik pora valandų per dieną, tiksliai identifikuotas išvykimo laikas – 85minutės. Tuo tarpu 92 minutes per dieną nustatoma išvykimas, kai jos nėra, ir 32 minutes išvykimas nėra fiksuojamas (4.2 lentelė).

4.2 lentelė. Išvykimo tikslumas, naudojant judesio jutiklius.

| n | p | Paklaida | Paklaida | Sutampa | Išvykę, tačiau nefiksuoja | Neišvykę, bet fiksuoja |
|----|------|----------|---------------|--------------|---------------------------|------------------------|
| 59 | 0,05 | 8,60% | 126 min/dieną | 85 min/dieną | 34 min/dieną | 92 min/dieną |

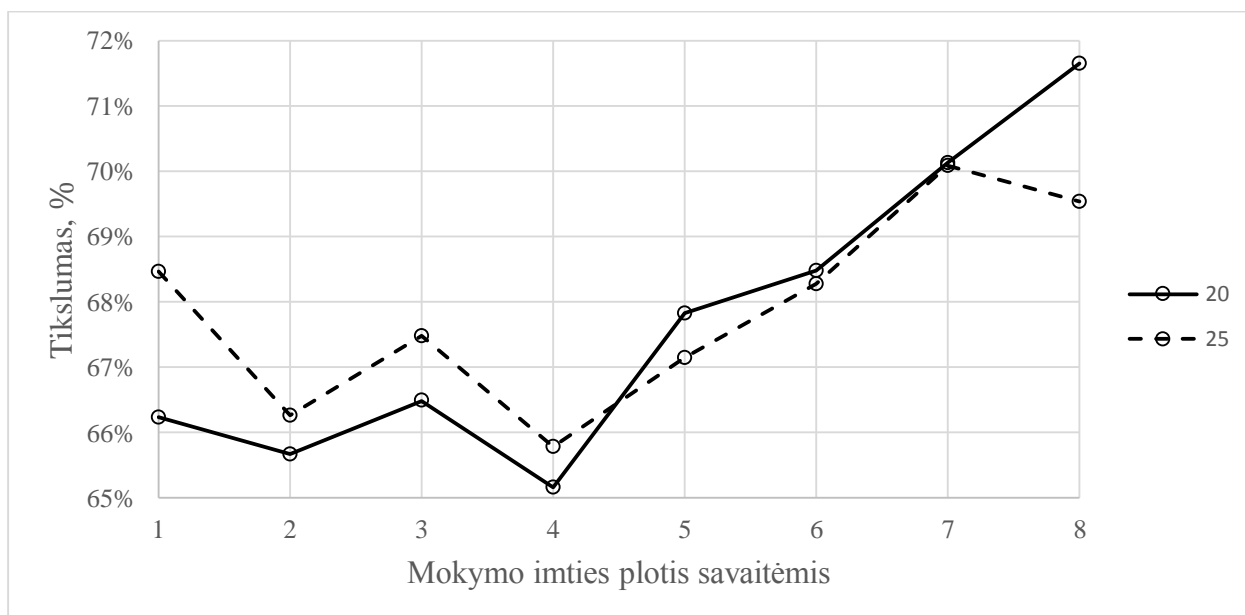
Iš duomenų nustatyto išvykimo tikslumas 48%, tokią paklaidą lemia trumpi išvykimo intervalai ir duomenų interpoliavimas.

4.2 Neuroninio tinklo prognozavimo tikslumas

Neuroninį tinklo mokymui buvo naudojamos trijų tipų duomenų imtys. Mokant neuroninį tinklą buvo tikrinami šie aspektai:

- papildomo įėjimo, atitinkančio savaitės dienas, įtaka prognozavimo tikslumui,
- papildomo įėjimo, atitinkančio darbo dienas, įtaka prognozavimo tikslumui,
- vieno mokymo ciklo atlikimo spartos priklausomybė nuo įėjimų ir paslėptų neuronų skaičiaus.

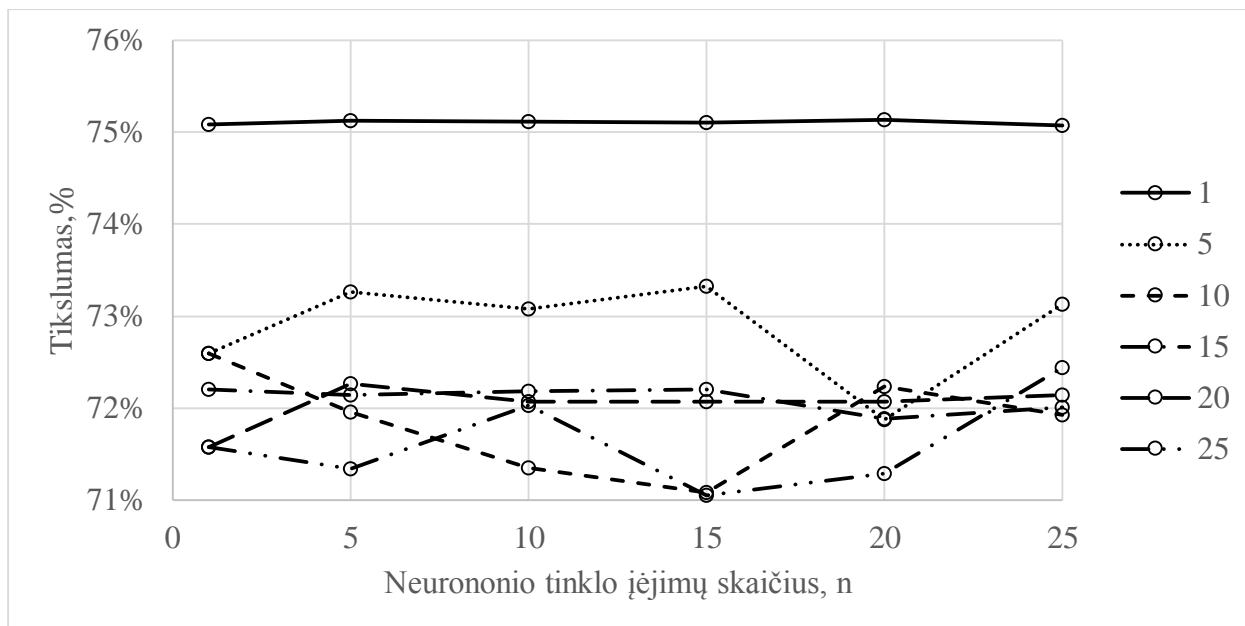
Prieš pradėdant sistemos mokymą su įvairiomis duomenų imtimis, buvo tiriama neuroninio tinklo prognozavimo tikslumo priklausomybė nuo tinklo mokymui skirtų duomenų imties pločio (4.1 pav.).



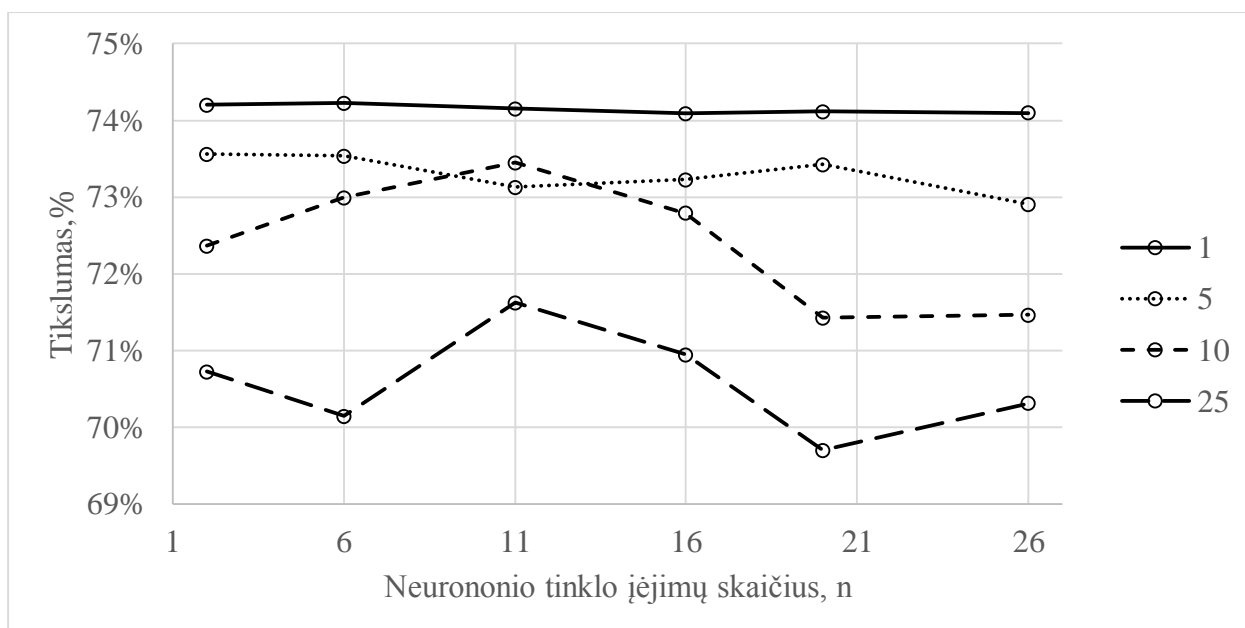
4.1 pav. Neuroninio tinklo prognozavimo tikslumo priklausomybė nuo tinklo mokymui skirtų duomenų imties

Tyrimas atliktas su neuroniniais tinklais, kurie paslėptame sluoksnyje turi 20 ir 25 neuronus, įėjimo vektoriaus plotis – 5. Rezultatai rodo, jog tinklo prognozavimo tikslumas ima didėti, kai duomenų imties plotis viršija keturias savaites. mokant tinklą su aštuonių savaičių duomenimis, tikslumas viršija 71%.

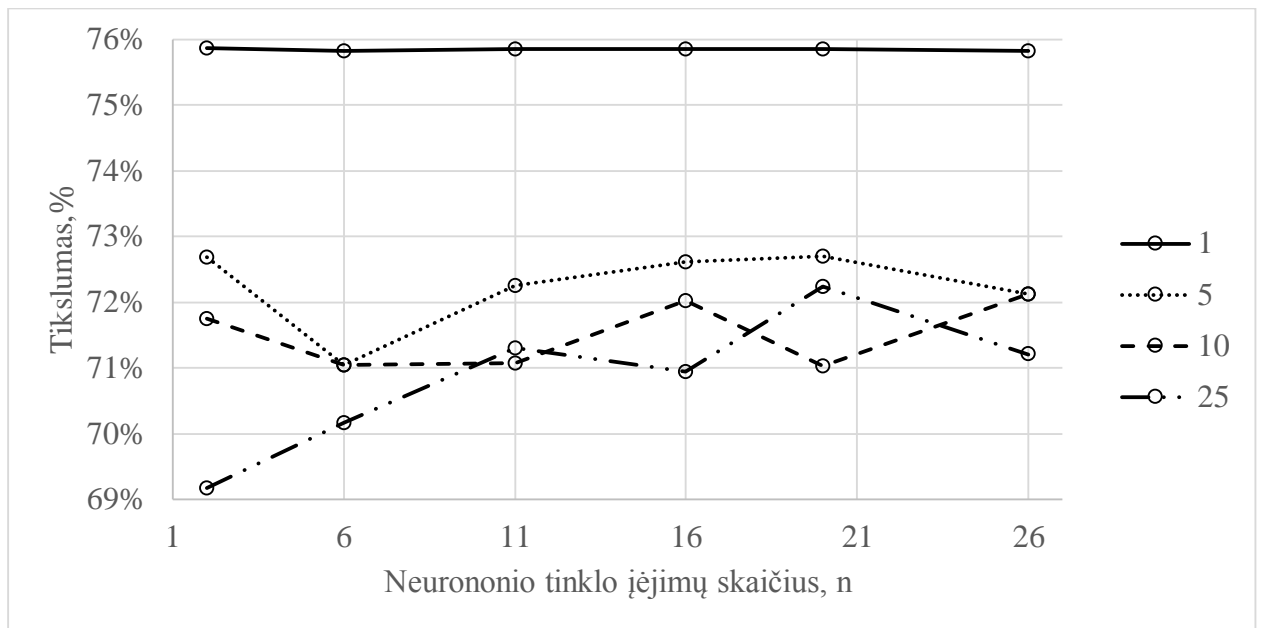
Buvo laikomasi nuostatos, jog visais atvejais neuroninis tinklas mokomas aštuonių savaičių duomenų imtimi. Neuroninis tinklas buvo mokomas prognozuoti užimtumo tipą dvi valandas į priekį. Prognozavimo tikslumui įvertinti buvo naudojama vienos savaitės duomenų imtis, kurios mokymo imtyje nebuvo.



4.2 pav. Tinklo prognozavimo tikslumo priklausomybė nuo įėjimų skaičiaus, kai tinklo mokymo duomenys suformuoti pagal gyventojų veiklą.



4.3 pav. Tinklo prognozavimo tikslumo priklausomybė nuo įėjimų skaičiaus, kai tinklo mokymo duomenys suformuoti pagal gyventojų veiklą ir pasitelkus papildomą, savaitės dienas nurodantį, įėjimą.



4.4 pav. Tinklo prognozavimo tikslumo priklausomybė nuo įėjimų skaičiaus, kai tinklo mokymo duomenys suformuoti pagal judesio jutiklių reikšmes ir pasitelkus papildomą, savaitės dienas nurodanti, įėjimą.

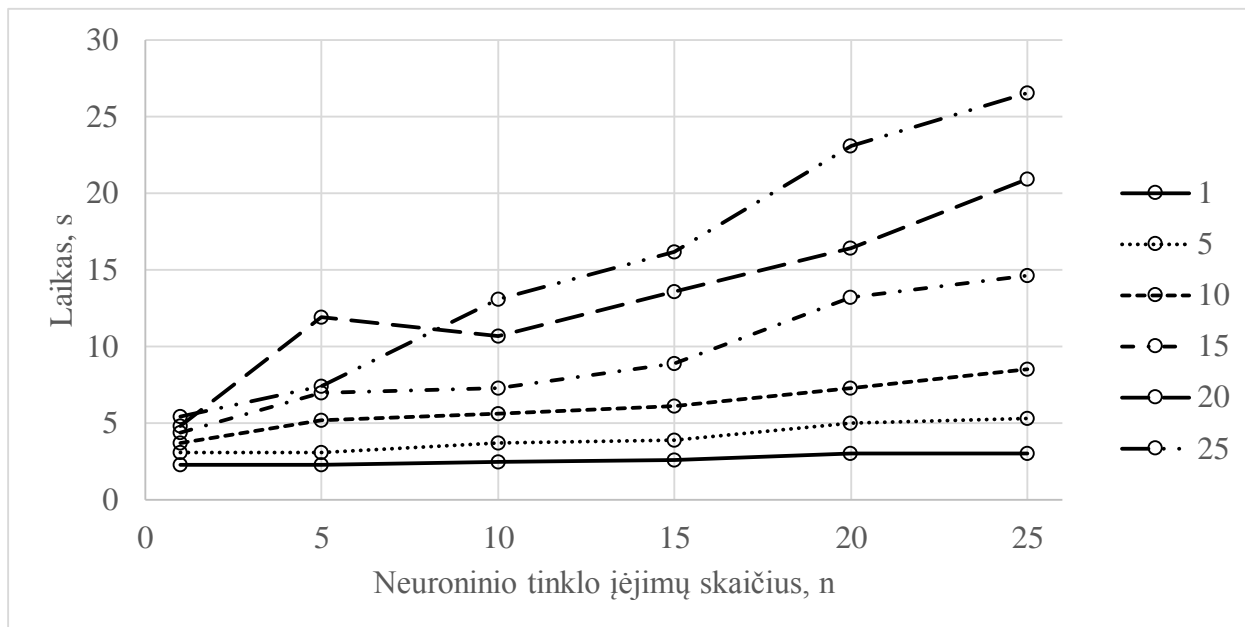
Gauti rezultatai rodo, jog geriausiai, nepriklausomai nuo duomenų imties (suformuota iš jutiklių ar gyventojų veiklos), bendrą trijų užimtumo tipų prognozavimą atlieka neuroninis tinklas, kuris paslėptame sluoksnyje turi vieną neuroną. Tinklą mokant su įvairiu neuroninio tinklo įėjimų skaičiumi n , bendras prognozavimo tikslumas svyruoja $\pm 0,2\%$. Geriausias $75,85 \pm 0,03\%$ (priklausomai nuo įėjimų skaičiaus) pasiekiamas, kai neuroninis tinklas mokomas su duomenimis, suformuotais iš judesio jutiklių reikšmių.

Fiksuoiant kiekvieno pastato užimtumo tipo prognozavimo tikslumą, tinklai mokyti be papildomo įėjimo (4.2 pav.), pastato užimtumą, kai miegama atpažino $92 \pm 2\%$ (priklausomai nuo įėjimų ir paslėptų neuronų skaičiaus), kai pastatas užimtas $90 \pm 3\%$, kai pastatas tuščias 0% tikslumu. Geriausiai išvykimą nustatė tinklai, kurie buvo mokyti su duomenimis, suformuotais iš jutiklių reikšmių, ir paslėptame sluoksnyje turėjo 25 paslėptus neuronus. Jie prognozavo $6 \pm 2\%$ tikslumu. Su 21 įėjimu pasiekiamas $95,6\%$ miego, $8,38\%$ tuščio ir $85,9\%$ užimto pastato tipų atpažinimo tikslumas.

4.5 paveiksle pavaizduota, kaip vidutinė vieno mokymo ciklo trukmė priklauso nuo įėjimo vektoriaus pločio, esant įvairiam paslėptų neuronų skaičiui. Laikas, reikalingas atlikti vienam mokymo ciklui, didėja, plečiantis įėjimo vektoriaus pločiui ir didėjant paslėptų neuronų skaičiui. Atlikti neuroninio tinklo prognozavimo tyrimą, mokant jį su duomenimis, suformuotais iš gyventojų veiklos (4.2 pav.), užtruko 15 h. Tyrimui pagreitinti buvo atsisakyta tirti neuroninius tinklus su 15 ir 20 paslėptų neuronų, nes šių neuroninių tinklų prognozavimo tikslumas mažesnis

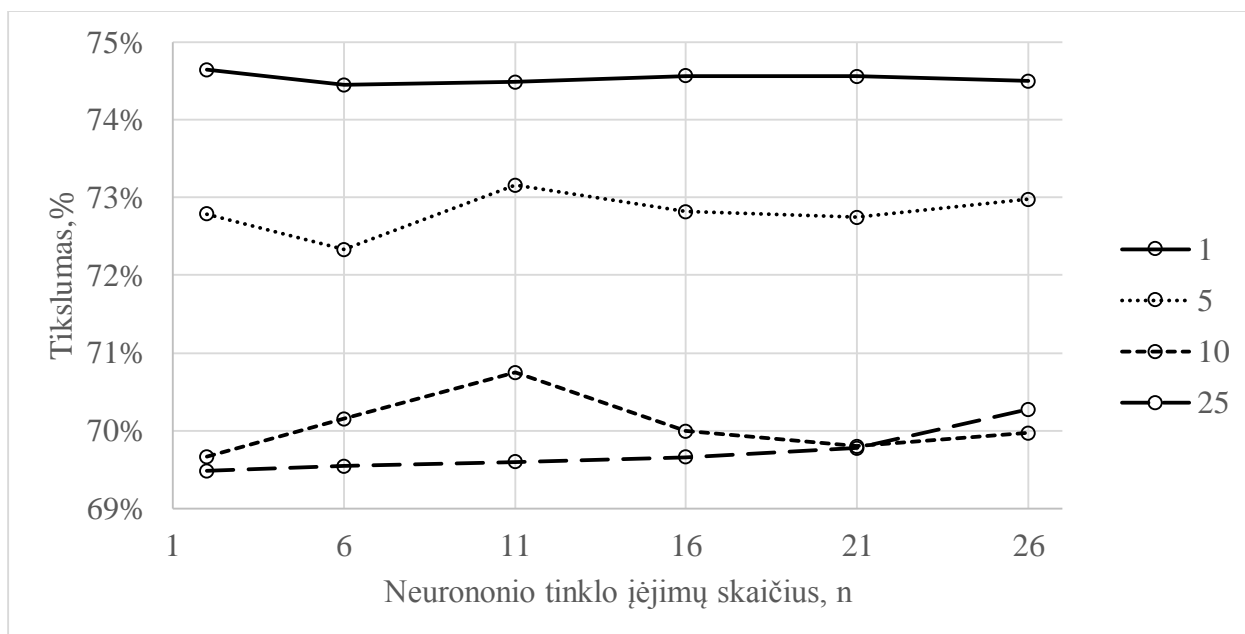
už tinklų turinčių 1, 2, 5 ir 10 paslėptų neuronų. Neuroninis tinklas su 25 paslėptais neuronais geriausiai atpažino, kada pastatas turi būti tuščias, todėl jo tikslumas bus tiriamas toliau.

Darant prielaidą, kad kiekvienas neuroninis tinklas apsimoko per j ciklą, mokymo laikas, netiriant tinklų su 15 ir 20 paslėptų neuronų, sutrumpėja 40%, tačiau realiai mokymo ciklų skaičius auga, didėjant paslėptų neuronų skaičiui ir visais mokymo atvejais būna skirtingas.

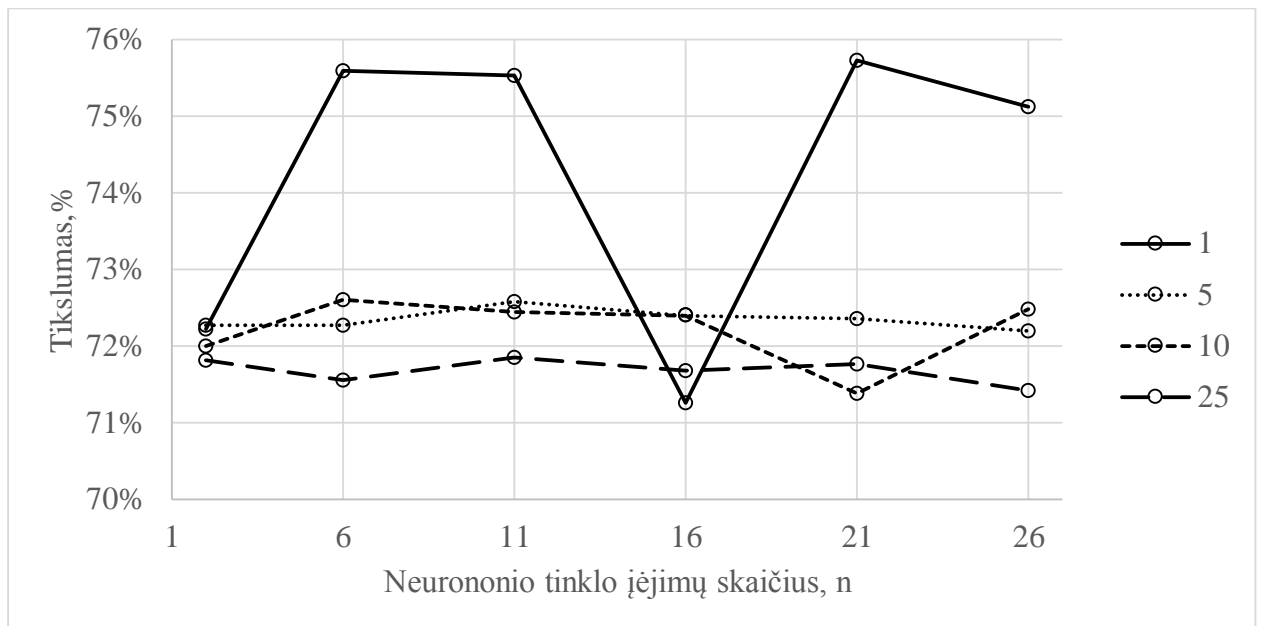


4.5 pav. Neuroninio tinklo vieno mokymo ciklo laiko priklausomybė nuo paslėptų neuronų skaičiaus

Neuroninius tinklus mokant su papildomu įėjimu, kuris atitinka darbo dienas ir savaitgalius, grafinė, neuroninio tinklų prognozavimo tikslumo, išraiška pateikta 4.6 ir 4.7 paveiksluose.



4.6 pav. Tinklo prognozavimo tikslumo priklausomybė nuo įėjimų skaičiaus, kai mokymui naudojami duomenys suformuoti iš gyventojų veiklos su papildomu, darbo dienas ir savaitgalius nurodančiu, įėjimu.

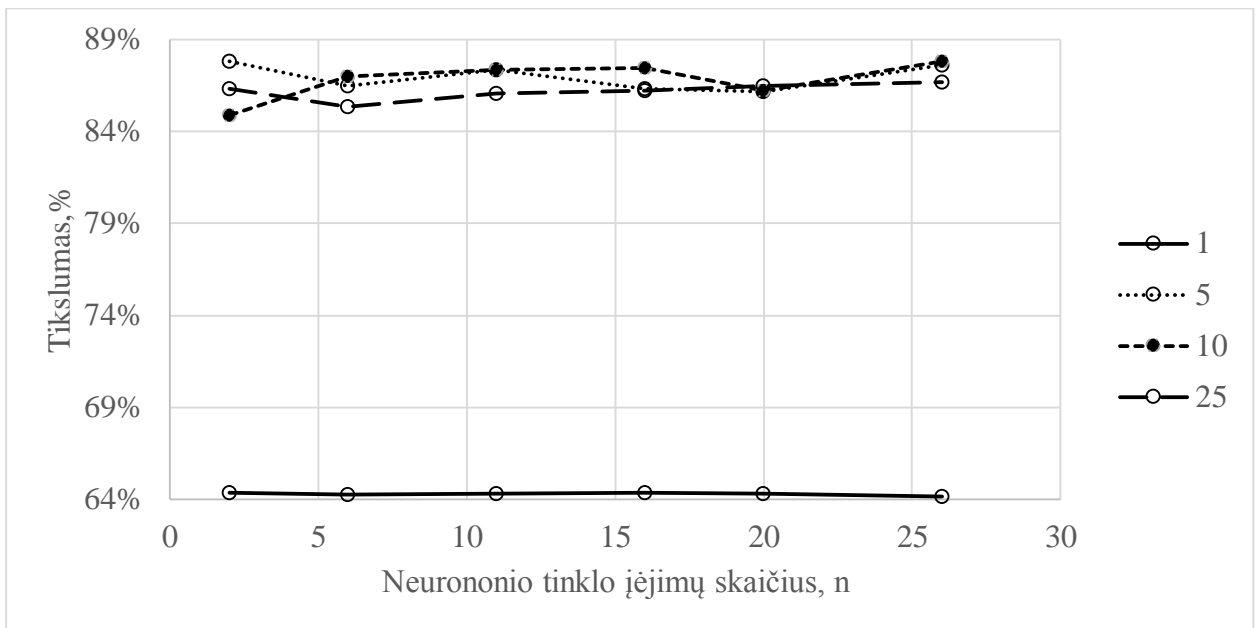


4.7 pav. Tinklo prognozavimo tikslumo priklausomybė nuo įėjimų skaičiaus, kai mokymui naudojami duomenys, suformuoti pagal judesio jutiklio reikšmes, su papildomu, darbo dienas ir savaitgalius nurodančiu, įėjimu.

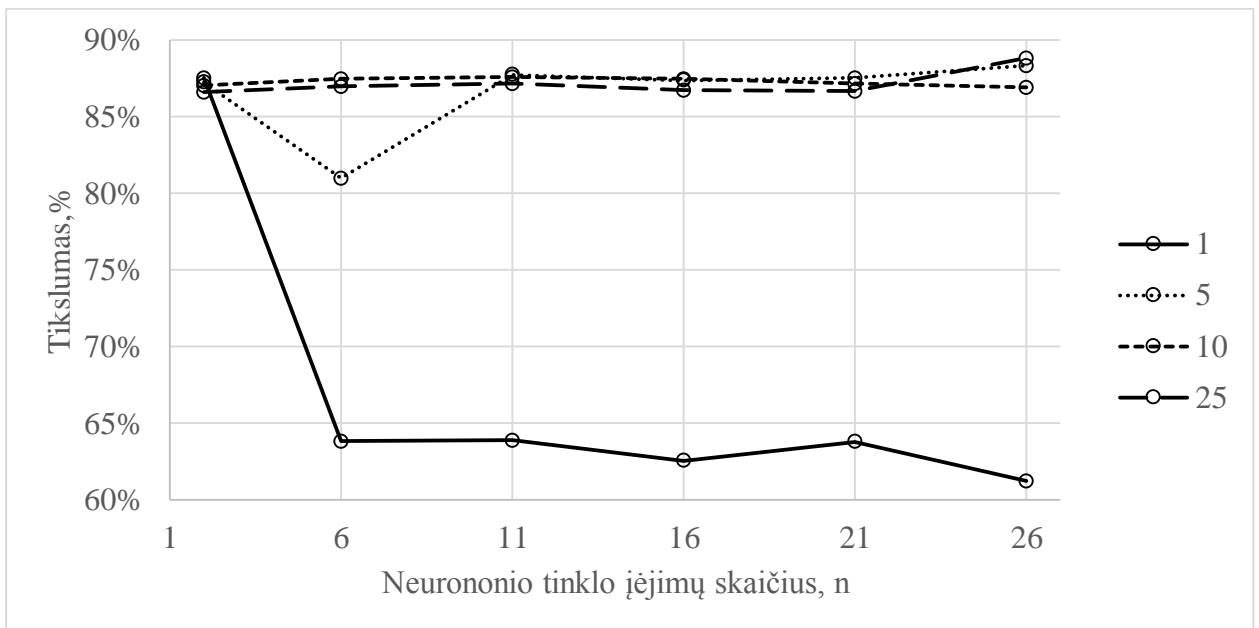
Rezultatai rodo, kad neuroniniui tinklui nurodžius darbo dienas ir savaitgalius atitinkantį papildomą įėjimą, tikslumas, lyginant su tinklais kuriems nurodoma savaitės dienos, keičiasi nežymiai $\pm 1\%$, tačiau visais atvejais tuščio pastato prognozavimo tikslumas 0%.

Tyrimo rezultatai parodė, kad nesistemingą ir trumpą neužimto pastato tipą neuroninis tinklas prognozuoja prastai (0 – 8,4%). Tinklo prognozavimo tikslumas padidėjo, kai mokymo ir tyrimo metu buvo naudojami duomenys, kuriems būdingas pastovus gyventojų išvykimas. Kadangi modeliuojant išvykimo duomenis buvo naudojami skirtingi darbo dienų ir savaitgalių išvykimo modeliai, su šiais duomenimis neuroninio tinklo tyrimas buvo vykdomas dviem etapais:

1. Vektorių N papildžius papildomu savaitės diena nurodančiu įėjimu (4.7 pav.);
2. Vektorių N papildžius papildomu darbo dienas ir savaitgalius atitinkančiu įėjimu (4.8 pav.).



4.7 pav. Tinklo prognozavimo tikslumo priklausomybė nuo įėjimų skaičiaus, kai mokymui naudojami sumodeliuoti duomenys, su papildomu, savaitės dienas nurodančiu, įėjimu.



4.8 pav. Tinklo prognozavimo tikslumo priklausomybė nuo įėjimų skaičiaus, kai mokymui naudojami sumodeliuoti duomenys, su papildomu, darbo dienas ir savaitgalius nurodančiu, įėjimu.

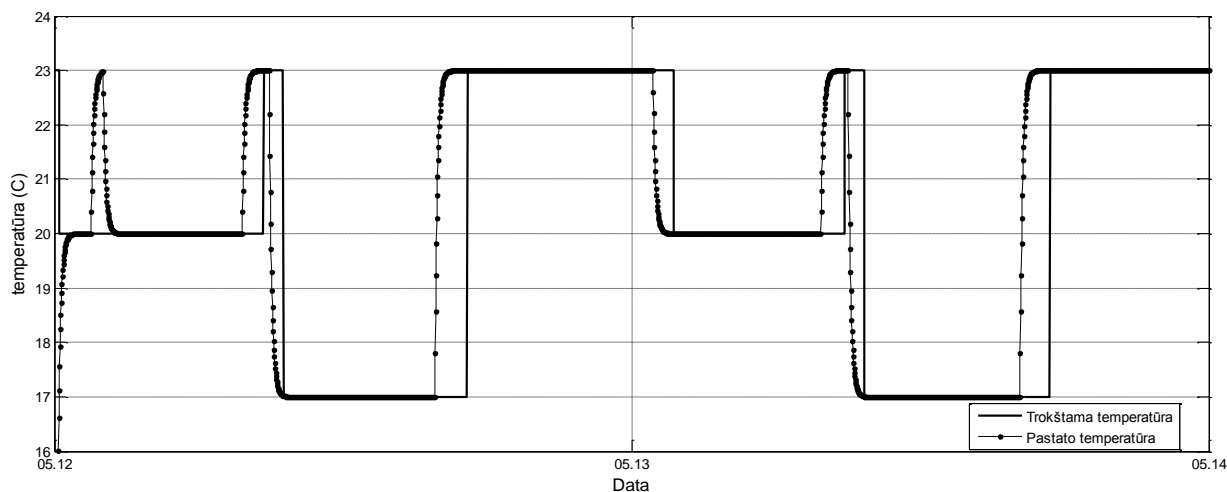
Rezultatai rodo, kad nepriklausomai nuo papildomo įėjimo, mokymo rezultatai yra panašūs. Neuroniniai tinklai, kurie paslėptame sluoksnyje turi vieną neuroną, neprognozuoja, kada pastatas bus tuščias, todėl jų bendras taiklumas yra apie 64%. Neuroniniai tinklai, turėdami 5, 10 ar 25 neuronus paslėptame sluoksnyje, atpažįsta visus tris pastato užimtumo tipus ir bendras prognozavimo tikslumas svyruoja nuo 84,85% iki 88,8%. Taip pat pastebėta, jog didėjant tuščio pastato prognozavimo tikslumui, užimto pastato prognozavimo tikslumas atitinkamai mažėja o gyventojų miego prognozei tai įtakos neturi.

4.3 Išmaniojo termostato veikimo tikslumas

Išmaniojo termostato pastato temperatūros reguliavimo grafinė išraiška pateikta 4.9 paveiksle. Termostatui nurodoma pradinė temperatūra $TempPID = 20\text{ }^{\circ}\text{C}$, laiką kurio reikia pastato temperatūrą pakelti vienu laipsniu – $LK = 10\text{ min}$, Išankstinis išjungimas $ANK = 0$. Prognozuoti naudojamas šešių įėjimų neuroninis tinklas, kur šeštasis neuronas nurodo savaitės dienas.

Šiluminei inercijai realizuoti, euristiškai parenkama pastato šilumos perdavimo funkcija. Perdavimo funkcija parenkama taip, jog tenkintų pradinę LK sąlygą, t.y. pastato temperatūra vienu laipsniu turi pakilti per 10 min. Į temperatūros kritimo greitį dėmesys nėra kreipiamas, nes tai nedaro įtakos sistemos tikslo vertinimui, kai nėra vertinama pastato šiluminė inercija, t.y. $ANK = 0$.

Įsijungus sistemai neuroniniam tinklui, keturias pirmas minutes, nurodomas klaidingas įėjimo vektorius N , kuriame bent vienas iš elementų yra 0. Dėl nulinių reikšmių įėjimo vektoriuje, sistema fiksuoja klaidingą užimtumą ir sukelia temperatūrą iki $23\text{ }^{\circ}\text{C}$.



4.9 pav. Išmaniojo termostato temperatūros reguliavimo lyginimas su realiu temperatūros poreikiu

Sistemos tikslumas vertinamas tuo, kaip tiksliai termostatas nustato užimto pastato temperatūros pradžią ir pabaigą. Lyginant siekiamos ir pastato $23\text{ }^{\circ}\text{C}$ laipsnių temperatūros intervalus, matyti, jog pastato temperatūra prasideda (vid. 32 min) ir baigiasi (vid. 45 min) anksčiau, tačiau bendras temperatūrų atitikimas yra 87%, t.y 77 min neužtikrinama maloni temperatūra.

REZULTATAI IR IŠVADOS

1. Pasiūlytu duomenų apdorojimo algoritmu pagal originalius duomenis buvo galima išskirti tris gyvenamųjų patalpų užimtumo tipus: kai jame miegama, kai jis yra tuščias ir kai jame aktyviai vykdoma veikla. Toks duomenų sugrupavimas yra tinkama pasirinkto neuroninio tinklo apmokymui.
2. Užimtumo tipas, kai gyventojai miega, atitiko 82,35% realaus miego. Vidutinis, vienos dienos, netiksliai identifikuotas laikas yra 156 minutės: 86 minutes veikla nustatoma, kada jos nėra, ir 72 minutes veikla nėra identifikuojama. Gyventojų išvykimas nustatomas 48% tikslumu. Tiksliai sistema identifikuoja 85 min, 95 min veikla identifikuojama, kai gyventojai nėra išvykę ir 32 min nėra identifikuojama.
3. Tiriant neuroninio tinklo prognozavimo tikslumą nustatyta, jog įėjimo vektoriaus plotis N nedarė įtakos prognozavimo tikslumui ir lėmė tik $\pm 1\%$ tikslumo svyravimą.
4. Didinant paslėptų neuronų skaičių, bendra prognozavimo paklaida didėjo, tačiau didesnis paslėptų neuronų skaičius lėmė tiksliau prognozuojamą gyventojų išvykimą.
5. Neuroninis tinklas, apmokytas su duomenimis, kuriems būdinga sistemingi pastato užimtumo tipai, prognozavo 88,8% tikslumu.
6. Išmanusis termostatas, reguliuodamas pastato šildymo sistemą, užtikrina 87% norimos šilumos atitikimą dviejų valandų bėgyje. Vidutiniškai termostatas 32 min per greitai pasiekia norimą temperatūrą ir 45 min per anksti nutraukia šildymą.
7. Termostatas fiksuoja bendrą viso pastato užimtumo tipą, todėl siūloma jog išmanusis termostatas valdytų ketureigį vandens pamaišymo vožtuvą, o pastato temperatūra būtų matuojama nuskaitant iš patalpų į rekuperatorių patenkančio oro temperatūrą.

INFORMACIJOS ŠALTINIŲ SĄRAŠAS

1. Energijos Informacijos Administracijos [žiūrėta 2015 05 13]. Prieiga per internetą:
<http://www.eia.gov/todayinenergy/detail.cfm?id=14011>
2. Energijos suvartojimas [žiūrėta 2015 05 13]. Prieiga per internetą:
http://www.eia.doe.gov/totalenergy/data/monthly/pdf/sec2_3.pdf
3. Namų ūkių energijos suvartojimas [žiūrėta 2015 05 13]. Prieiga per internetą:
http://www.eia.doe.gov/totalenergy/data/annual/pdf/sec2_19.pdf
4. SCOTT J., PreHeat: Controlling Home Heating Using Occupancy Prediction, *UbiComp'11*, September 17–21, 2011, Beijing, China.
5. MOZER M.C., The Neural Network House: An Environment that Adapts to its Inhabitants, *American Association for Artificial Intelligence Spring Symposium on Intelligent Environments*, Menlo Park, 1998, p. 110-114.
6. CAMPO E., Le concept ergdom, *Annales du Bâtiment et des Travaux Publics*, 2003, p. 1270-9840
7. JIAKANG L., The Smart Thermostat: Using Occupancy Sensors to Save Energy in Homes, *SenSys'10*, November 3–5, Zurich, Switzerland, 2010.
8. KRUMM J., BRUNCH A., Learning Time-Based Presence Probabilities, *Pervasive Computing Lecture Notes in Computer Science*, vol 6696, 2011, p. 79-96
9. ELLIS C., EarlyOff: Using House Cooling Rates To Save Energy, *Buildsys'12*, Toronto, ON, Canada, November 6, 2012.
10. MOZER M. C., The Neurothermostat: predictive Optimal control of Resident Heating Systems, *Advances in Neural Information Processing Systems 9*, Cambridge, p. 953 – 959, 1997.
11. GUPTA M., STEPHEN S., Adding GPS-Control to Traditional Thermostats: An Exploration of Potential Energy Savings and Design Challenges, *Pervasive Computing Lecture Notes in Computer Science*, Volume 5538, p. 95-114, 2009,
12. DZEMYDA G., Daugiamačių duomenų vizualizavimo metodai, p. 108 – 118, Vilnius 2008
13. VISIOLI A., Practical PID control, *Advances in industrial control*, p. 18 – 24, 2006
14. Vašingtono valstijos universiteto duomenų bazė, [žiūrėta 2015 m. gegužės 8 d.]. Prieiga per internetą : <http://casas.wsu.edu/datasets/>

PRIEDAI

Priedas 1. M matricos formavimo Matlab išėities kodas

```
laikas = strcat(Var1,Var2);
datenum(strcat(Var1(56562,1),Var2(56562,1)), 'yyyy-mm-ddHH:MM:SS')

M1=zeros(length(Laikas),31);
true = strncmp('ON', busena,2);
false = strncmp('OFF', busena,3);
j=1;
while j<=length(davikliai)
x = strncmp(davikliai(j,1), ID,4);
ML1=x&true;
MM1=ones(length(Laikas),1);
MM1=MM1.*ML1;

ML1=x&false;
MM2=ones(length(Laikas),1)*2;
MM2=MM2.*ML1;
A=MM2+MM1;
m=2; i=1126;
while i<=length(Laikas)
if A(i,1)~=0;
m=A(i,1);
end
if m==1
A(i,1)=1;
else A(i,1)=0;
end
i=i+1;

end
M1(1:end,j)=A;
j=j+1;
end
```

Priedas 2. Duomenų interpoliavimo Matlab išėities kodas

```
To=Laikas(sav(pr,1):sav(pb,2),1);
To=[datenum(datestr(Laikas(sav(pr,1),1), 'dd-mmm-yyyy'), 'dd-mmm-yyyy'); To;...
datenum(datestr(Laikas(sav(pb,2)+1,1), 'dd-mmm-yyyy'), 'dd-mmm-yyyy')];
Do=M(sav(pr,1):sav(pb,2),:);
zeros=zeros(1,31);
Do1 =[zeros; Do ;zeros];
Is=[0; Isvyk(sav(pr,1):sav(pb,2),1); 0];
Mieg_and=[0; Miegas_and(sav(pr,1):sav(pb,2),1); 0];

[To1, ia, ic]=unique(To);
Do2=(Do1(ia,:));
```

```

Is1=Is(ia,1);
Mieg1=Mieg_and(ia,1);

Ti=(min(To1):1/24/60:max(To1));
Di = interp1(To1,Do2,Ti,'nearest');

Isi = interp1(To1,Is1,Ti,'nearest');
Mieg_andi=interp1(To1,Mieg1,Ti,'nearest');

```

Priedas 3. Duomenų išlyginimo Matlab išėties kodas

```

i=1; ist=1; k=30; n=59; p=0.05; e=0;
while i<=(length(D22)-n)
    if i>k
        ist=sum(D22(i-k:i,1));
    end

    if ist~=0
        m=sum(D22(i:i+n,1));
        if (m/(n+1))>=p
            D22(i,1)=1;
        else D22(i,1)=0;
        end
    end
    i=i+1;
end
D22=bitxor(1, D22)*2;% naudojama jeigu identifikuojamas isvykimas

```

Priedas 4. N ir T imčių sudarymo Matlab išėties kodas

```

Wd=weekday(Ti); i=1;
while i<=length(Ti);
    if Wd(i,1)~=1&&Wd(i,1)~=7
        Wd(i,1)=1;
    else
        Wd(i,1)=2;
    end
    i=i+1;
end
Ti=datetime(datestr(datetime(Ti),'HH:MM:SS'),'HH:MM:SS');
Dq=Isi+Mieg_andi;
Dq_D=Isvyk_D+Mieg_D;
m=1; i=1;
x1=i+nn+A;%A prognozavimo i ateiti laikas min

while i+nn+A<=length(Ti);

```

```

for k=1:nn
    N(k,i)=Ti(m,1);
    m=m+1;
end
Wdn(i)=Wd(i+nn+A,1);
i=i+1;
m=i;
end

N=[Wdn;N];
% Uu=[Uu ; Ui(1:end-1,1)];
T1= Dq_D==0;
T2=Dq_D==1;
T3=Dq_D==2;

TN=[T1(x1:end,1)';T2(x1:end,1)';T3(x1:end,1)'];
TN=ones(3,length(TN)).*TN;

```

Priedas 5. Išmaniojo termostato Matlab išeities kodas

```

pr=datenum('2015-05-12','yyyy-mm-dd');
pb=datenum('2015-05-14','yyyy-mm-dd');

N=zeros(11,1);
Temp=[23;20;17];
info=zeros(3,120);
LK=10/1440;
ANK=3/1440;
q=0; a=0;i=1;

s = tf('s');
G = 3*(s+4)/(s*(s+25)*(s+3));
T = feedback(G,1);

YTemp=0;
TempPID=20;
int=1/1440;
for t=pr:int:pb

    Wd=weekday(t);
    Ti=datenum(datestr(datenum(t),'HH:MM:SS'),'HH:MM:SS');
    if Wd<6
        N(1,1)=1;
    else
        N(1,1)=2;
    end
    N(2:10,1)=N(3:11,1);
    N(11,1)=Ti;

```

```

    y=net(N);
    for j=1:3
    if y(j,1)>q
        q=y(j,1);
        c=j;
    end
    end

b(1,i)=c;
Tb(1,i)=t+1/12;

if i>=2
    if b(1,i-1)==2&&b(1,i)==1//b(1,i-1)==3&&b(1,i)==1// b(1,i-1)==3&&b(1,i)==2
        a=1;
    end
    if b(1,i-1)==1&&b(1,i)==2//b(1,i-1)==1&&b(1,i)==3// b(1,i-1)==2&&b(1,i)==3
        a=2;
    end
end
info(:,1:end-1)=info(:,2:end);

if a~=0
    info(1,120)=a;
    info(2,120)=Temp(c,1);
    info(3,120)=t+1/12;
else
    info(:,120)=[0;0;0];
end

if sum(sum(info))~=0

[ row col ] = find(info);
Prio=info(row(1,1),col(1,1));
TempX=info(row(2,1),col(1,1));
Lx=info(row(3,1),col(1,1));

dTemp = TempX-YTemp(end,1);
Ly = LK*dTemp;
Li=t;
dL=Lx-t;

if Prio==1 && Ly >=dL && YTemp(end,1)>=16
    TempPID = TempX;

    else if Prio==2 && Li>=(Lx-ANK ) && YTemp(end,1)>=16
        TempPID = TempX;
    end
end

end
temperat(i,1)=TempPID;
u(i,1)=TempPID;

```

```
tt(i,1)= i-1;  
  
if length(u)>=2  
[YTemp,tf,x] = lsim(T,u,tt);  
end  
i=i+1;  
a=0; q=0;  
end
```