LATVIJAS UNIVERSITĀTE
FIZIKAS UN MATEMĀTIKAS FAKULTĀTE
MATEMĀTIKAS NODAĻA

NESTRIKTAS KLASTERIZĀCIJAS METODES BALSTĪTAS UZ F-TRANSFORMĒTIEM DATIEM

BAKALAURA DARBS

Autors: **Mihails Anufrijevs**
St. apliecības Nr.: ma11033
Darba vadītāja: profesore, Dr. Math., Svetlana Asmuss

Rīga, 2015

Anotācija

Darbs ir veltīts F-transformācijām un to pielietojumam datu klasterizācijas uzdevumā. Darbā tika apskatītas nultās un pirmās pakāpes F-transformācijas, nestriktas klasterizācijas metodes, kā arī validācijas indeksi un nestriktas klasifikācijas metodes. Apskatītas datu transformācijas un klasterizācijas metodes tika izmēģinātas uz uzģenerētām laikrindām, kas apraksta tīkla trafika plūsmu. Tika veikta iegūto rezultātu analīze un interpretācija.

Atslēgas vārdi: F-transformācija, datu transformācija, nestriktas klasterizācijas metodes, FCM algoritms, laikrindu klasterizācija, laikrindu klasifikācija.

Abstract

The Paper is devoted to F-transform and its implementation in data clustering problem. Zero and first degree F-transforms, fuzzy clustering methods as well as validation index and fuzzy classification methods were reviewed in this Paper. Reviewed data transforming and clustering methods have been used on generated time series, which simulate network traffic flow. The analysis and interpretation of the obtained results have been made.

Key words: F- transformation, data transformation, fuzzy clustering methods, FCM algorithm, time series clustering, time series classification.

Saturs

IEVADS **5**

1. F-TRANSFORMĀCIJAS **6**

1.1. F-transformācijas jēdziens 6

1.2. Nepārtraukta F-transformācija 7

1.3. Diskrēta F-transformācija 7

1.4. F^1-transformācija 10

2. KLASTERIZĀCIJAS UN KLASIFIKĀCIJAS ALGORITMI **12**

2.1. Nestriktas klasterizācijas jēdziens 12

2.2. Nestriktais c-vidējo algoritms 13

2.3. Validācijas indeksi 15

2.4. Klasifikācijas uzdevums 16

3. PRAKTISKĀ DAĻA **17**

3.1. DDoS uzbrukumi 17

3.2. Datu simulācijas process 17

3.3. Datu transformācijas 20

4. TRANSFORMĒTU DATU KLASTERIZĀCIJAS ANALĪZE **22**

4.1. Klasterizācija balstīta uz F-transformācijas koeficientu kombinācijām . . . 22

4.2. Klasterizācija balstīta uz F^1-transformācijas otro koeficientu 27

4.3. Secinājumi 28

LITERATŪRAS SARAKSTS **30**

PIELIKUMI **31**

Apzīmējumi

FCM - nestriktais c-vidējo algoritms (*Fuzzy C-Means*, FCM)

PC - sadales koeficients (*Partition Coefficient*)

MPC - modificētais sadales koeficients (*Modified Partition Coefficient*)

FS - Fukujama un Sugeno (*Fukuyama and Sugeno*) validācijas indekss

XB - Ksī un Beni (*Xie and Beni*) validācijas indekss

$\mathbb{R}$ - reālu skaitļu telpa

F^0 - nultās pakāpes F-transformācija

F^1 - pirmās pakāpes F-transformācija

c^0 - F^1-transformācijas pirmās komponentes vektors

c^1 - F^1-transformācijas otrās komponentes vektors

A_i - i-tā bāzes funkcija, $i = 1, .., m$

t_i - i-tais mezglu punkts, $i = 1, .., m$

U- nestrikta atbilstības matrica

u_{ij} - j-tā elementa piederības pakāpe pie i-tā klastera

μ - svara eksponente, kura raksturo "izplūdumu" starp objektiem

d - metrikas funkcija

d_{ij} - attālums starp x_j un c_i

C - klasteru centroīdu vektors

c_i - i-ta klastera centroīds

k - klasteru skaits

$\sharp$ - vektoru konkatenācijas simbols

y^T - vektora y transponētais vektors

IEVADS

Klasteru analīze ir instruments, kas palīdz atklāt nezināmu objektu kopas struktūru. Klasterizācijas mērķis ir sagrupēt objektus pa klasteriem tā, lai katrs no klasteriem sastāvētu pēc iespējas no lidzīgākiem objektiem, bet objekti no dažādiem klasteriem būtiski atšķirtos. Savukārt, klasifikācijas uzdevums ir noteikt jaunā objekta piederību kādam no jau atdalītajiem objektu klasteriem.

Klasterizācijas algoritmus iespējams sadalīt divās grupās, striktā klasteru analīze un nestriktā klasteru analīze. Darbā tiks izmantotas nestriktās klasterizācijas metodes.

Klasterizācijas uzdevums ir svarīgs priekš daudzām dažādām zinātņu nozarēm. Tā, piemēram, klasterizācija ir pielietojama un svarīga DDoS uzbrukumu kostatēšanā. Analizējot tīkla trafika plūsmu mēs sastopamies ar milzīgiem datu apjomiem. Kas, savukārt, rada jaunu uzdevumu, samazināt datu apjomu tā, lai pēc iespējas mazāk zaudēt informācijas par datu struktūru. Tātad šī darba galvenais mērķis ir apskatīt F-transformācijas un pielietot tos datu klasterizācijā.

Darbs sastāv no četram nodaļām. Darba pirmajā nodaļā ir apskatītas ar F- transformācijām saistītie jēdzieni tajā skaitā kopas nestriktais sadalījums, kopas nestriktas apakškopas. Otrajā nodaļā ir aprakstītas izmantotās metodes, nestriktais klasterizācijas algoritms, validācijas indeksi un nestriktais klasifikācijas algoritms. Trešajā nodaļā ir aprakstīti ģenerētie dati un ir veiktas datu transformācijas. Ceturtajā nodaļā ir veikta transformētu datu klasterizācija un tas rezultātu analīze. Nobeigums apkopo padarīto darbu, kā arī ir piedāvāti varianti tālakajiem pētījumiem. Pielikumā ir ievietots programmas kods, kas bija izmantots darbā.

1. F-TRANSFORMĀCIJAS

1.1. F-transformācijas jēdziens

F-transformācijas jēdziens balstās uz nestrikto sadalījumu. Lai runātu par nestrikto sadalījumu, vispirms ir jāapskata kopas nestriktas apakškopas jēdziens.

Definīcija.[1] Par kopas X nestriktu apakškopu sauc attēlojumu
$A : X \longmapsto [0, 1]$.

Tagad, izmantojot kopas nestriktas apakškopas, mēs varam definēt kopas nestrikto sadalījumu.

Definīcija.[2] Ņem, fiksētus mezglus $t_1, ..., t_m$ no $[a, b]$ tādus, ka

$$a = t_1 < t_2 < .. < t_{m-1} < t_m = b, m \geq 2. \tag{1}$$

Saka, ka nestriktās kopas $A_1, ..., A_m$ ir bāzes funkcijas, kas veido intervāla $[a, b]$ nestrikto sadalījumu, ja tās apmierina šādus nosacījumus:

1. $A_i : [a, b] \longmapsto [0, 1], A_i(t_i) = 1, i = 1, .., m$;
2. $A_i(t) = 0$, ja $t \in (t_{i-1}, t_{i+1}), i = 1, .., m$, kur apzīmējumu vienādībai pieņemsim, ka $t_0 = a$ un $t_{m+1} = b$;
3. A_i ir nepārtraukta, $, i = 1, .., m$;
4. A_i ir monotoni augoša intervālā $[t_{i-1}, t_i], i = 2, ..., m$, un A_i ir monotoni dilstoša intervālā $[t_i, t_{i+1}], i = 1, .., m - 1$;
5. visiem $t \in [a, b]$
$$\sum_{i=1}^{m} A_i(t) = 1. \tag{2}$$

Šajā darbā apskatīsim trijstūraveida sadalījumu. Vienmērīgo trijstūraveida sadalījumu, kur mezgli tiek fiksēti vienādā attālumā $t_i = t_{i-1} + h, i = 1, .., m$, var redzēt 1.1. attēlā. Un 1.2. attēlā ir parādīts nevienmērīgais trijstūraveida sadalījums.

Sekojošās formulas aprakstā nestrikto sadalījumu intervālam $[t_1, t_m]$ ar m trijstūraveida bāzes funkcijām:

$$A_1(t) = \begin{cases} 1 - \frac{t-t_1}{h_1}, & \text{ja } t \in [t_1, t_2], \\ 0, & \text{citādi,} \end{cases}$$

$$A_i(t) = \begin{cases} \frac{t-t_{i-1}}{h_{i-1}}, & \text{ja } t \in [t_{i-1}, t_i], \\ 1 - \frac{t-t_i}{h_i}, & \text{ja } t \in [t_i, t_{i+1}], \\ 0, & \text{citādi,} \end{cases} \qquad i = 2, ..., m-1,$$

$$A_m(t) = \begin{cases} \frac{t-t_{m-1}}{h_{m-1}}, & \text{ja } t \in [t_{m-1}, x_m], \\ 0, & \text{citādi,} \end{cases}$$

kur $h_i = t_{i+1} - t_i, i = 1, .., m-1$.[2]

1.2. Nepārtraukta F-transformācija

Nepārtraukta F-transformācijas ir attēlojums, kas pārveido nepārtrauktu funkciju f m-dimensiju vektorā. Savukārt, inversa F-transformācija pārveido m-dimensiju vektoru par nepārtrauktu funkciju, kas varētu būt apskatīta kā sakotnējas funkcijas f aproksimācija.

Definīcija.[3] Pieņemsim, ka nestriktais sadalījums intervālam $[a, b]$ ir uzdots ar bāzes funkcijām $A_1, ..., A_m, m \geq 2$, un ir paņemta patvaļīga un nepārtraukta funkcija $f : [a, b] \longmapsto \mathbb{R}$. Tad reālo skaitļu m-dimensiju kortežs $(F_1, ..., F_m)$, kas uzdots ar formulu

$$F_i = \frac{\int_a^b f(t)A_i(t)dt}{\int_a^b A_i(t)dt}, i = 1, .., m, \tag{3}$$

ir tiešā F-transformācija no f attiecībā pret doto nestrikto sadalījumu. $F_1, ..., F_m$ ir komponentes F-transformācijai no f.

Definīcija.[3] Pieņemsim, ka $f : [a, b] \longmapsto \mathbb{R}$ ir patvaļīga nepārtraukta funkcija un $F_m[f] = (F_1, ..., F_m)$ ir tiešā F-transformācija funkcijai f attiecībā pret bāzes funkcijām $A_1, ..., A_m$. Tad funkciju $f_{F,m}$, kas uzdota intervālā $[a, b]$ ar formulu

$$f_{F,m}(t) = \sum_{i=1}^{m} F_i A_i(t) \tag{4}$$

sauc par inverso F-transformāciju no f.

1.3. Diskrēta F-transformācija

Pieņemsim, ka intervāls $[a, b]$ ir fiksēta universāla kopa, $t_1, ..., t_m$ ir fiksēti intervāla $[a, b]$ sadalījuma mezgli, tādi ka ir spēkā (1). Pieņemsim, ka punkti $p_1, ..., p_l$ ir fiksēti punkti no intervāla $[a, b]$, tādi ka $m \geq 2, l > m$. Pieņemsim, ka $A_1, ..., A_m$ ir fiksētas bāzes funkcijas, kuras uzdod nestrikto sadalījumu intervālam $[a, b]$. Pieņemsim, ka kopa

$P_l = \{p_1, ..., p_l\}$ ir pietiekami blīva (*sufficiently dense*) attiecībā pret sadalījumu $A_1, ..., A_m$, t.i.

$$\forall i \in 1, .., m \exists j \in 1, .., l : A_i(p_j) > 0.$$

Apskatīsim telpu V_l, kas sastāv no funkcijam ar reālam vērtībam, kuras ir definētas kopā P_l, t.i.

$$V_l = \{f : P_l \longmapsto R\}.$$

Ja priekš katra $f \in V_l$ ievest apzīmējumu $f_j = f(p_j), j = 1, ..., l$, tad kopā V_l varētu būt reprezentēta kā kopa no visiem iespējamiem reālas vērtības l-dimensionāliem vektoriem.[2]

Definīcija.[2] Pieņemsim, ka ir dota funkcija $f \in V_l$ un $A_1, ..., A_m$ ir fiksētas bāzes funkcijas, $m < l$. Tad saka, ka m-dimensiju reālu skaitļu kortežs $(F_1, ..., F_m)$ ir funkcijas f diskrēta F-transformācija attiecībā pret $A_1, ..., A_m$, ja izpildās

$$F_i = \frac{\sum_{j=1}^{l} f(p_j) A_i(p_j)}{\sum_{j=1}^{l} A_i(p_j)}, i = 1, .., m. \tag{5}$$

Apzīmēsim funkcijas f diskrēto F-transformāciju attiecībā pret $A_1, ..., A_m$ ar

$$F_m[f] = (F_1, ..., F_m).$$

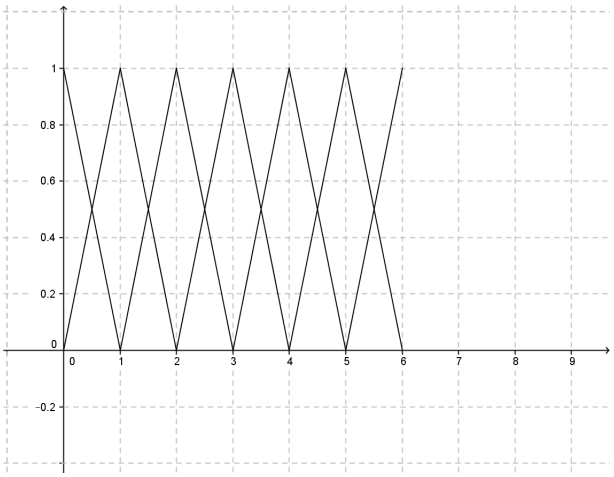
Funkcija f varētu būt tuvināti rekonstruēta, izmantojot inversās formulas no tās F-transformācijas.

Definīcija.[2] Pieņemsim, ka ir dota funkcija f un $F_m[f] = (F_1, ..., F_m)$ ir funkcijas f diskrēta F-transformācija attiecībā pret $A_1, ..., A_m$. Tad funkciju

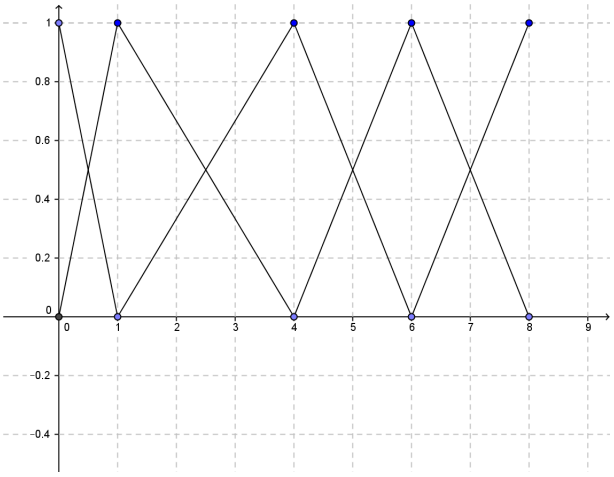
$$f_{F,m}(p_j) = \sum_{i=1}^{m} F_i A_i(p_j), j = 1, ..., l, \tag{6}$$

kura ir definēta uz tās pašas kopas P_l, sauc par inverso diskrēto F-transformāciju.

Tālāk darbā sauksim F-transformāciju par F^0-transformāciju, jo tiks izmantota arī pirmās pakāpes F-transformācija, kuru, savukārt, apzīmēsim ar F^1. Iepriekšējos paragrāfos priekš nepārtrauktas un diskrētas F-transformācijas bija izmantoti vienādi apzīmējumi, bet tā kā darbā būs izmantota tikai diskrēta transformācija, tad tas neradīs pārpratumus.



1.1 att.: **Intervāla [0,6] vienmērīgais trijstūrveidas sadalījums**



1.2 att.: **Intervāla [0,8] nevienmērīgais trijstūraveida sadalījums**

1.4. F^1-transformācija

Nepārtraukta F^1-transformācija

Pieņemsim, ka nestriktais sadalījums intervālam $[a, b]$ ir uzdots ar bāzes funkcijām $A_1, ..., A_m, m \geq 2$, un ir paņemta patvaļīga nepārtraukta funkcija $f : [a, b] \longmapsto R$. Tad m-dimensiju kortežs $(F_1^1, ..., F_m^1)$ ir funkcijas f F^1-transformācija attiecībā pret $A_1, ..., A_m$, ja katram $i = 1, .., m$ ir spēkā

$$F_i^1(t) = c_i^0 + c_i^1(t - t_i), t \in [t_{i-1}, t_{i+1}],$$

kur koeficienti c_i^0, c_i^1 ir uzdoti ar formulām

$$c_i^0 = \frac{\int_{t_{i-1}}^{t_{i+1}} f(t)A_i(t)dt}{\int_{t_{i-1}}^{t_{i+1}} A_i(t)dt}, \tag{7}$$

$$c_i^1 = \frac{\int_{t_{i-1}}^{t_{i+1}} f(t)(t - t_i)A_i(t)dt}{\int_{t_{i-1}}^{t_{i+1}} (t - t_i)^2 A_i(t)dt}. \tag{8}$$

Diskrētā F^1-transformācija

Pieņemsim, ka intervāls $[a, b]$ ir fiksēta universāla kopa, $t_1, ..., t_m$ ir fiksēti mezgli. Ir dota funkcija $f \in V_l (V_l$, kuru ir definēta līdzīgā veidā, kā F^0-transformācijas gadījumā) un $A_1, ..., A_m$ ir fiksētas bāzes funkcijas, $m < l$. Tad saka, ka m-dimensiju kortežs $F_m^1[f] = (F_1^1, ..., F_m^1)$ ir funkcijas f F^1-transformācija attiecībā pret $A_1, ..., A_m$, ja katram $i = 1, .., m$, un $p_j \in P_l$ ir spēkā

$$F_i^1(p_j) = c_i^0 + c_i^1(p_j - t_i), \tag{9}$$

kur koeficienti c_i^0, c_i^1 ir uzdoti ar formulām

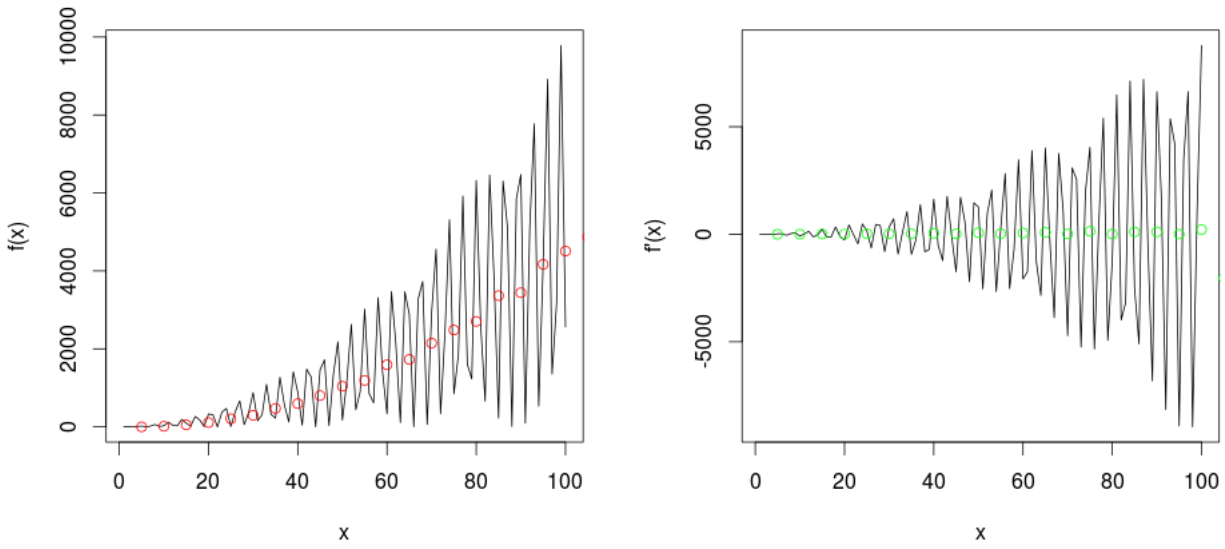
$$c_i^0 = \frac{\sum_{j'=1}^{l} f(p_{j'})A_i(p_{j'})}{\sum_{j'=1}^{l} A_i(p_{j'})}, \tag{10}$$

$$c_i^1 = \frac{\sum_{j'=1}^{l} f(p_{j'})(p_{j'} - t_i)A_i(p_{j'})}{\sum_{j'=1}^{l} (p_{j'} - t_i)^2 A_i(p_{j'})}. \tag{11}$$

No dotām formulām ir redzams, ka koeficients c_i^0 pilnīgi sakrīt ar augstāk aprakstītās F^0-transformāciju, kas sniedz informāciju par sākuma funkciju f. Savukārt, otrais koeficients c_i^1 dod informāciju par funkcijas f pirmo atvasinājumu punktā t_i. Informācija,kas bija

izmantota šajā paragrafā paņemta no avota [8].

Piemērs Apskatīsim F-transformācijas piemēru. Pieņemsim, ka ir dota funkcija $f(t) = t^2 sin^2(t)$, bet mēs izmantosim tikai tās diskrētas vērtības $f(p_j) = p_j^2 sin^2(p_j)$, punktos $p_j = j$, kur $j = 0, 1, 2, .., 100$. Tad funkcijas atvasinājuma vektors ir $f'(p_j) = 2p_j sin(p_j)(sin(p_j) - p_j cos(p_j))$. Kreisajā attēlā(1.3. att.) ir funkcija un tas transformācijas komponentes c_i^0 , labajā attēlā ir funkcijas pirmais atvasinājums un transformācijas koeficienti c_i^1, kur intervāla sadalījumu veido punkti $t_i = 5(i-1), i = 1, .., 21$.



1.3 att.: **F^1-transformācijas komponēntes c^0 un c^1 attiecīgi.**

2. KLASTERIZĀCIJAS UN KLASIFIKĀCIJAS ALGORITMI

2.1. Nestriktas klasterizācijas jēdziens

Klasteru analīzes, jeb vienkārši klasterizācijas, uzdevums ir datu sagrupēšana attiecīgajās klasēs. Klasteru analīze ir pielietojama daudzās zinātņu nozārēs, tādas, kā ķīmija, bioloģija, medicīna, socioloģija un t.t. Pastāv dažādas klasterizācijas metodes un algoritmi, kas var sniegt dažādu klasteru sadalījumu uz tiem pašiem datiem. Klasterizācijas metodes iespējams sadalīt striktās un nestirktās klasterizācijas metodēs. Pēc striktas klasterizācijas metodem tiek uzskatīts, ka pētāmais objekts pieder tikai un vienīgi vienam klasterim. Pie striktām klasterizācijas metodēm attiecās k-vidējo algoritms. Darbā izmantosim nestriktas metodes, kas atļauj objekta piederību pie dažādiem klasteriem, bet katram klasterim ar savu piederības pakāpi.

Apskatīsim dažas definīcijas un ieviesim apzīmējumus, kuras mēs izmantosim rakstot par klasterizācijas algoritmiem.

Definīcija. Funkciju $d : X \times X \longrightarrow \mathbb{R}$ sauc par metriku, ja visiem $x, y, z \in X$ ir spēkā:

1. $d(x, y) \geq 0$,
2. $d(x, y) = 0 \Leftrightarrow x = y$,
3. $d(x, y) = d(y, x)$,
4. $d(x, z) \leq d(x, y) + d(y, z)$.

Šajā darbā mēs izmantosim Eiklīda metriku, kura varētu būt uzrakstīta ar sekojošu formulu

$$d(x, y) = \sqrt{(x - y)^T (x - y)},$$

kur x un y ir vektori. Šajā nodaļa mēs klasterizēsim kopu $X = \{x_1, .., x_n\}$, kur x_i, $i = 1, .., n$ ir vektori. Mēs pieņemsim, ka klasteru skaits ir k.

Definīcija. Matricu $U = [u_{ij}]_{k \times n}$ sauc par nestrikto atbilstības matricu, ja tas elementi u_{ij} satur informāciju par kopas X j-tā objekta piederību i-tajam klasterim un apmierina šādus nosacījumus:

1. $u_{ij} \in [0, 1], i = 1, ..., k, j = 1, ..., n$,
2. $\sum_{i=1}^{k} u_{ij} = 1, j = 1, ..., n$,
3. $\sum_{j=1}^{n} u_{ij} \in (0, 1), i = 1, ..., k$.

2.2. Nestriktais c-vidējo algoritms

Nestriktais c-vidējo algoritms (FCM) ir vispazīstamākais un ir bieži pielietotais no nestriktiem klasterizācijas algoritmiem. FCM ir iteratīvais algoritms ar nepieciešamo nosacījumu minimizēt mērķa funkciju J_{FCM}. Kura ir izteikta formā

$$J_{FCM}(X,U,C) = \sum_{i=1}^{k}\sum_{j=1}^{n} u_{ij}^{\mu} d_{ij}^2, \tag{12}$$

kur $\mu > 1$ ir svara eksponente, kura raksturo "izplūdumu" starp klateru robežām. Šeit ir n datu skaits, k ir klasteru skaits, u_{ij} ir objekta x_j piederības pakāpe pie i-ta klastera un d_{ij} ir attālums starp x_j un c_i (c_i ir i-tā klastera centroīds),t.i.

$$d_{ij} = \sqrt{(x_i - c_j)^T (x_i - c_j)}.$$

Lai iegūtu nestrikto sadali, parametru μ izvēlas lielāku par vienīnieku. Gadījumā, kad $\mu = 1$ FCM, algoritms veic strikto klasterizāciju, jo tad formula (9) pilnīgi atbilst k-vidējo algoritma mērķa funkcijai. Procesa sākumā izvēlas kādus klasteru centrus un mēģina minimizēt mērķa funkciju, atjauninot u_{ij} un klasteru centrus ar šādām formulām:

$$u_{ij} = \frac{d_{ij}^{-\frac{2}{\mu-1}}}{\sum_{i'=1}^{k} d_{i'j}^{-\frac{2}{\mu-1}}}, i = 1,...,k, j = 1,...n, \tag{13}$$

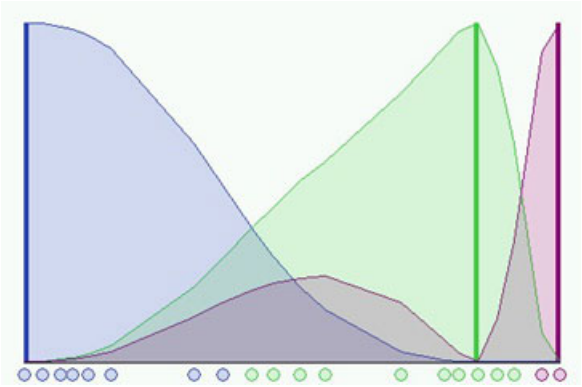
$$c_i = \frac{\sum_{j=1}^{n} u_{ij}^{\mu} x_j}{\sum_{j=1}^{n} u_{ij}^{\mu}}, i = 1,...,k. \tag{14}$$

Soļus (13) un (14) atkārto kamēr nebūs sasniegta vēlama precizitāte. Informācija par FCM algoritmu ir paņemta no avota [5].

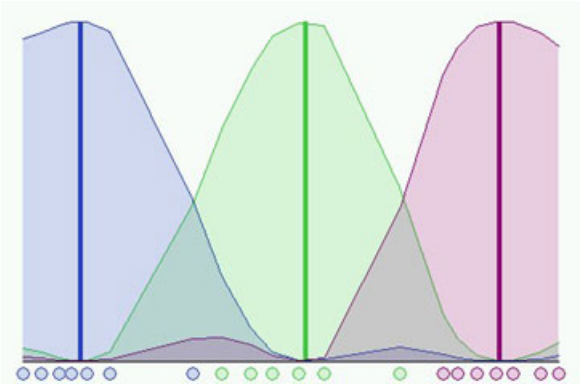
Piemērs[9]. Apskatīsim FCM algoritma darbību. Pieņemsim, ka ir viendimensionāls gadījums ar datiem apjomā 20, un ir zināms, ka tos var sagrupēt 3 klasteros. Attēlā 2.1a ir parādīts datu piederības pakāpes katram klasterim. Datu krāsa mainās atkarībā no piederības pie klasteriem.

Attēlā parādīts sākotnējais stāvoklis, kur nestriktais sadalījums ir atkarīgs no klasteru izvietojuma. Tā kā algoritms nav izdarījis nevienu soli, tad klasteri vēl nav labi atdalīti. Pieņemot $\mu = 2$ un precizitāti 0.3 algoritms beidz darbību 8. solī. Rezultāti redzami attēlā 2.1b.

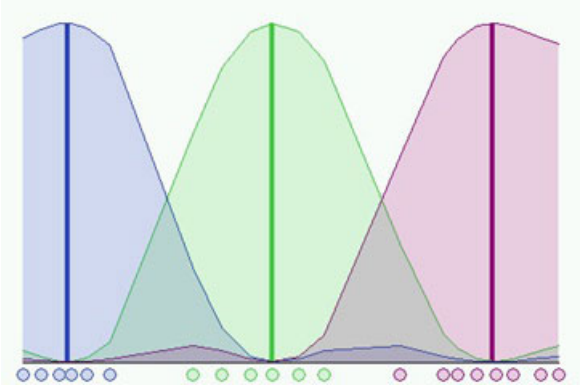
Palielinot precizitāti līdz 0.01 ir iespējams uzlabot rezulātus. Algoritms apstāsies tikai 37. solī, rezultātu iespējams redzēt attēlā 2.1c



(a) **Stāvoklis pirms FCM algoritms ir palaists**



(b) **FCM algoritma rezultāts 8. solī**



(c) **FCM algoritma rezultāts 37. solī**

2.1 att.

2.3. Validācijas indeksi

Klasteru analīzes uzdevums ir noteikt grupas ar līdzīgām pazīmēm. Tomēr lielākai daļai no klasterizācijas algoritmiem, lai darboties, ir jāzina klasteru skaits. Bet, vairumā gadījumu, lietotājam nav nekādas iepriekšējas zināšanas par to, cik klasēs mēs sadalam datu kopu. Gadījumā, kad atdalītu klasteru skaits ir lielāks par faktisku klasteru skaitu, viens vai vairāki klasteri varētu būt sasmalcināti. Vai arī gadījumā, kad faktiskais skaits ir lielāks par iegūto klasteru skaitu, viens vai vairāki klasteri nav atpazīti kā atsevišķi klasteri. Tāpēc klasteru pareizo skaitu noteikšana ir svarīgs uzdevums. Parasti to sauc par klasteru validāciju. Kad datu kopas sadalījums ir pabeigts, ar klasterizācijas algoritmu palīdzību, tad validācijas funkcija var palīdzēt validēt, novērtēt klasterizāciju.

Pastāv dažādi nestriktās klasterizācijas validācijas indeksi. Validācijas indeksus iespējams sadalīt uz divām grupām. Rēķinot pirmās grupas indeksus ir vajadzīga tikai elementu atbilstības matrica. Otrā veida indeksiem ir vajadzīga gan elementu atbilstības matrica, gan arī paši dati.

Apskatīsim dažus indeksus, kas tika izmantotas klasteru validācijai [6]:

- Sadales koeficients (*Partition Coefficient*, PC)

$$PC(k) = \frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{n} u_{ij}^2, \tag{15}$$

kur $\frac{1}{k} \leq PC(k) \leq 1$. Labākus klasterizācijas rezultātus mēs sasniegsim pie optimālā klasteru skaita k^*, kas tiek reķināts pie nosacījuma, ka

$$PC(k^*) = max_{2\leq k\leq n-1} PC(k).$$

- Modificētais sadales koeficients (*Modified Partition Coefficient* , MPC)

$$MPC(k) = 1 - \frac{k}{k-1}(1 - PC(k)). \tag{16}$$

Indeksa vērtība ir apgabalā $[0, 1]$. Optimālu klasteru skaitu k^* meklē izmantojot nosacījumu, ka

$$MPC(k^*) = max_{2\leq k\leq n-1} MPC(k).$$

- Fukujama un Sugeno (*Fukuyama and Sugeno*, FS) indekss ir definēts ar formulu

$$FS(c) = J_{FCM}(X, U, C) - \sum_{j=1}^{n} u_{ij}^{\mu} d(c_i, \overline{c})^2 =$$

$$= \sum_{i=1}^{k}\sum_{j=1}^{n} u_{ij}^{\mu} d(x_j, c_i)^2 - \sum_{i=1}^{k}\sum_{j=1}^{n} u_{ij}^{\mu} d(c_i, \overline{c})^2, \tag{17}$$

kur $C = (c_1, .., c_k)$ ir klasteru centru vektors un $\overline{c} = \frac{1}{k}\sum_{i=1}^{k} c_i$. Pirmā summa $J_{FCM}(X, U, C)$ ir FCM algoritma mērķa funkcija, kas raksturo klasteru kompaktumu. Savukārt, summa $\sum_{j=1}^{n} u_{ij}^{\mu} d(c_i, \overline{c})^2$ raksturo klasteru atdalamību. Optimālais klasteru skaits k^* ir atrodams aprēķinot $FS(k^*) = min_{2 \le k \le n-1} FS(k)$

- Ksī un Beni (*Xie and Beni* , XB) indekss ir definēts ar formulu

$$XB(c) = \frac{J_{FCM}(X, U, C)}{n \cdot min_{i,j} d(c_i, c_j)^2} =$$

$$= \frac{\sum_{i=1}^{k}\sum_{j=1}^{n} u_{ij}^{\mu} d(x_j, c_i)^2}{n \cdot min_{i,j} d(c_i, c_j)^2}. \tag{18}$$

Ksī un Beni piedāvātais indekss balstās uz divām īpašībām: kompaktums un atdalāmība. XB indeksa skaitītājs raksturo nestrikta sadalījuma kompaktumu, bet dalītājs raksturo klasteru atdalīšanu savā starpā. Optimālais klasteru skaits k^* ir atrodams aprēķinot $XB(k^*) = min_{2 \le k \le n-1} XB(k)$.

2.4. Klasifikācijas uzdevums

Klasifikācijai ir uzdevums noteikt kāda jauna objekta piederības pakāpi pie jau izdalītājām klasēm. Darbā būs izmantots tuvāko prototipu klasifikācijas algoritms.

Klasifikācijas algoritms izmanto informāciju par klasteru prototipiem, kas bija noteiktā klasterizācijas posmā. Vektors $C = (c_1, .., c_k)$ ir klasterizācijas procesā iegūtie klasteru centroīdi. Tuvāko prototipu klasifikācijas algoritmā centroīdus izmanto vektoru U^* aprēķināšanai, kas apraksta jauna objekta t^* piederību pie jau izdalītajiem klasteriem. Vektoru U^* aprēķināšanai izmanto klasterizācijas algoritmu atjaunināšanas formulas. Tā kā darbā ir izmantots FCM, tad formula pēc kuras rēķināsim U^* ir šāda

$$u_{ij}^{*} = \frac{d_{ij}^{-\frac{2}{\mu-1}}}{\sum_{i'=1}^{k} d_{i'j}^{-\frac{2}{\mu-1}}}, i = 1, ..., k, j = 1, ...n.$$

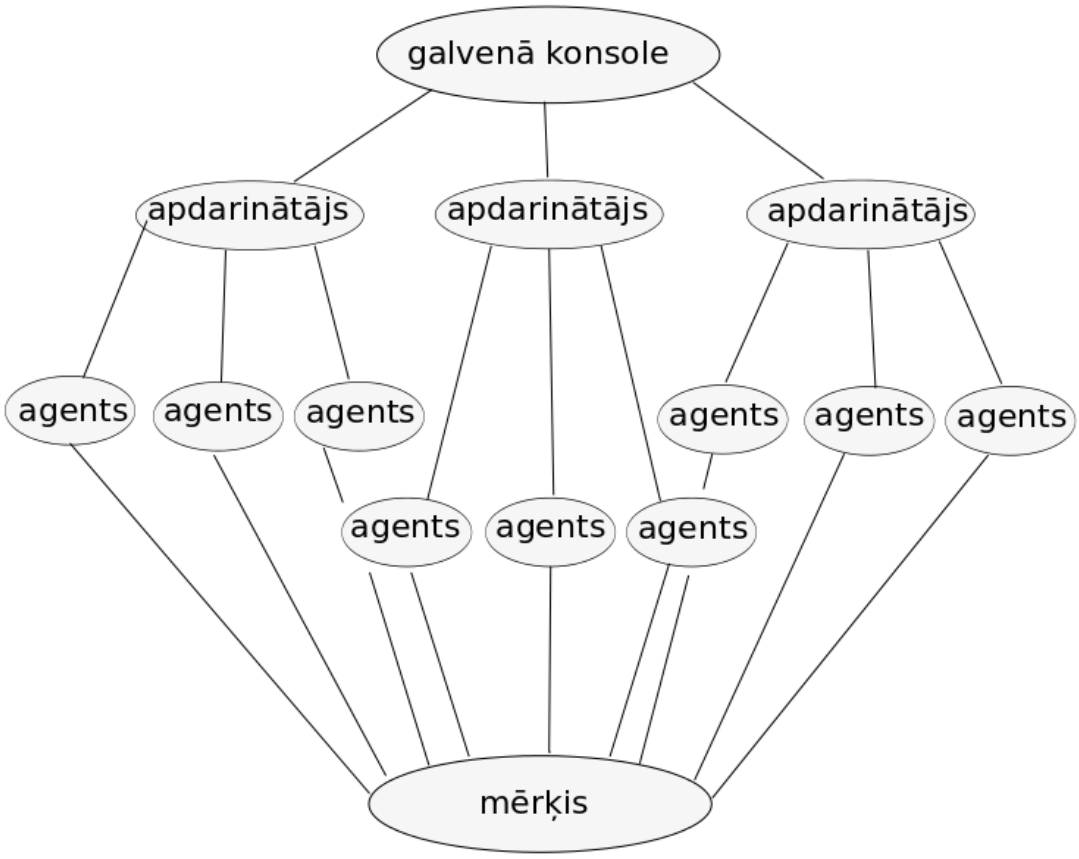
Vairāk par nestriktām klasifikācijas metodēm iespējam izlasīt avotā [7].

3. PRAKTISKĀ DAĻA

3.1. DDoS uzbrukumi

DDoS uzbrukums (*Distributed Deniel of Service Attack*) ir viens no svarīgākiem tīklu draudiem. Uzbrukuma mērķis ir paralizēt servera darbību. DDoS uzbrukumi var ātri izterēt tīkla resursus vai servera jaudu, kas novedīs pie nespējas sniegt pakalpojumus.

Darbā apskatīsim lietojumslāņa pārpludināšanas uzbrukumus, kas tiek raksturoti ar pieprasījumu skaita palielināšanu.



3.1 att.: **DDoS uzbrukuma piemērs ar trīslīmeņu arhitektūru**

Pastāv dažādas uzbrukuma arhitektūras, viena no izplatītākajām ir trīslīmeņu arhitektūra. Galvenā konsole padod signālu par uzbrukuma iesākšanu datoriem, ko sauc par apdarinātājiem, kas padod šo signālu tālāk agentiem. Tiešā uzbrucēju lomā šeit uzstājas agentu tīkls. Agentu tīkls var būt izveidots no dažiem desmitiem līdz tūkstošiem datoru. Parasti agenti ir neitrāli datori, kas bija inficēti ar kaitīgajām programmām. Programmas strādā fona režīmā un nepārtraukti sūta pieprasījumus uz serveri.

3.2. Datu simulācijas process

Analīzējot lokālā tīkla trafika plūsmu, izmantosim uzģenerētas laikrindas. Datu ģenerēšanai un apstrādei izmantosim programmu R. Detalizēta veida laikrindas ģenerēšanas

metode un laikrindas saistība ar lokāla tīkla plūsmas trafiku ir aprakstītas darbā [4]. Mēs izmantosim tājā aprakstītu laikrindu ģenerēšanas algoritmu un programmas kodu.

Darbā būs uzģenerētas divu veidu uzbrukumi. Pirmā veida uzbrukums raksturojas ar trafika plūsmas vērtību momentālo pacelšanu līdz kaut kādam konstantam līmenim. Otrā veida uzbrukums raksturosies ar trafika plūsmas lineāru palielināšanu kaut kāda laika intervālā.

Analīzei tika uzģenerētas 15 laikrindu kopas ar dažādu uzvedību. Katrā kopā sastāv no 100 parstāvjiem, tātad kopumā tika uzģenerētās 1500 laikrindas ar garumu 10000 laika vienības (Uzģenerētu datu paraugi ir apskatāmi 3.2. attēlā).

A1: Raksturo vienmērīgu, konstantu trafikas plūsmu, bez uzbrukuma.

A2: Raksturo trafikas plūsmu ar augošo trendu, bez uzbrukuma.

A3: Raksturo trafikas plūsmu ar dilstošo trendu, bez uzbrukuma.

B1: Raksturo vienmērīgu, konstantu trafikas plūsmu, pievienojot konstantu uzbrukumu, kas sākas no laika $\tau_1 \in (3000, 4000)$ un beidzas laikā $\tau_2 \in (6000, 7000)$.

B2: Raksturo trafikas plūsmu ar augošo trendu, pievienojot konstantu uzbrukumu, kas sākas no laika $\tau_1 \in (3000, 4000)$ un beidzas laikā $\tau_2 \in (6000, 7000)$.

B3: Raksturo trafikas plūsmu ar dilstošo trendu, pievienojot konstantu uzbrukumu, kas sākas no laika $\tau_1 \in (3000, 4000)$ un beidzas laikā $\tau_2 \in (6000, 7000)$.

C1: Raksturo vienmērīgu, konstantu trafikas plūsmu, pievienojot augošo uzbrukumu, kas sākas no laika $\tau_1 \in (3000, 4000)$ un beidzas laikā $\tau_2 \in (6000, 7000)$.

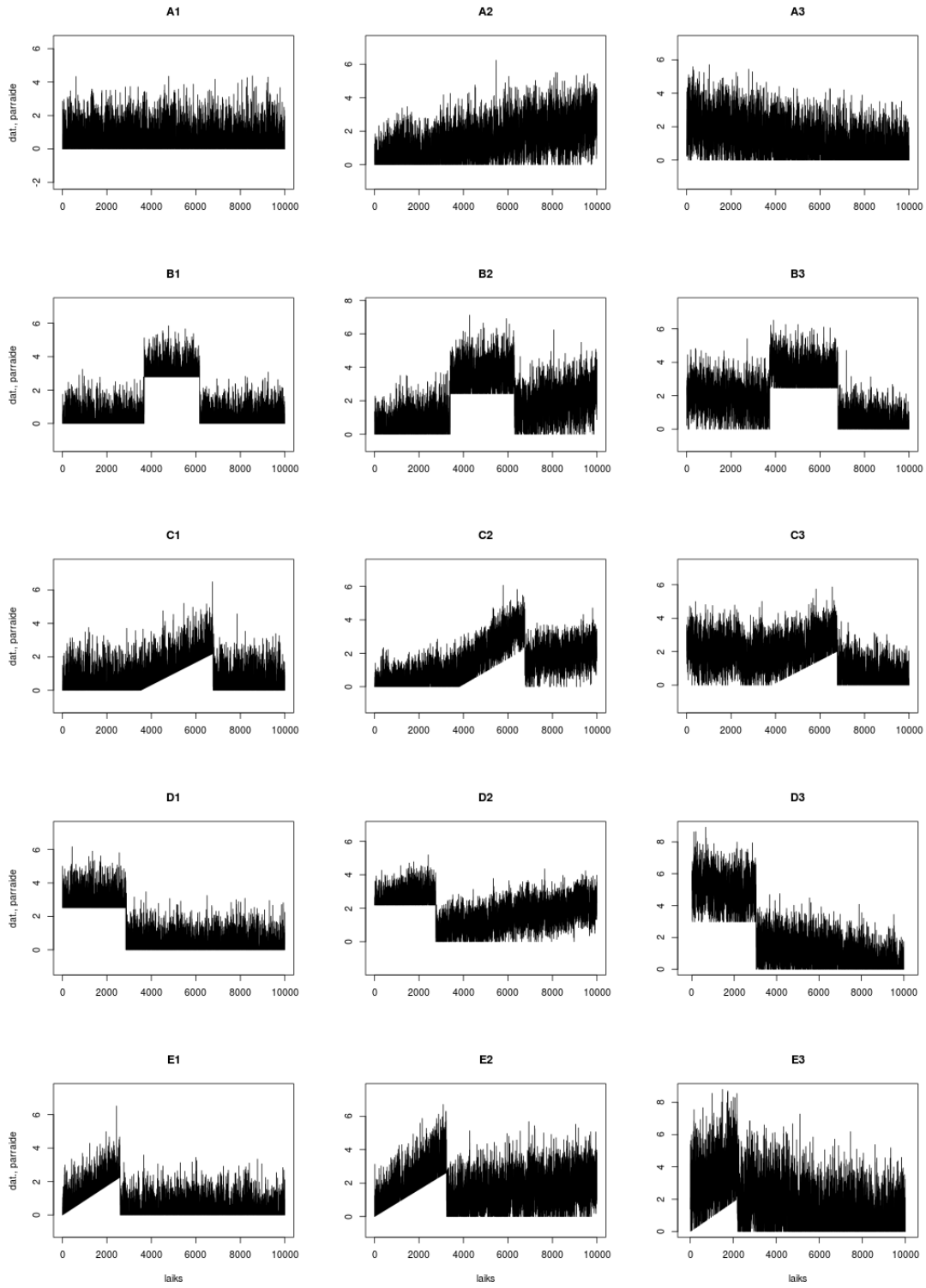
C2: Raksturo trafikas plūsmu ar augošo trendu,pievienojot augošo uzbrukumu, kas sākas no laika $\tau_1 \in (3000, 4000)$ un beidzas laikā $\tau_2 \in (6000, 7000)$.

C3: Raksturo trafikas plūsmu ar dilstošo trendu, pievienojot augošo uzbrukumu, kas sākas no laika $\tau_1 \in (3000, 4000)$ un beidzas laikā $\tau_2 \in (6000, 7000)$.

D1: Raksturo vienmērīgu, konstantu trafikas plūsmu, pievienojot konstantu uzbrukumu, kas sākas laikā $\tau_1 = 0$ un beidzas laikā $\tau_2 \in (6000, 7000)$.

D2: Raksturo trafikas plūsmu ar augošo trendu,pievienojot konstantu uzbrukumu, kas sākas laikā $\tau_1 = 0$ un beidzas laikā $\tau_2 \in (6000, 7000)$.

D3: Raksturo trafikas plūsmu ar dilstošo trendu, pievienojot konstantu uzbrukumu, kas sākas laikā $\tau_1 = 0$ un beidzas laikā $\tau_2 \in (6000, 7000)$.



3.2 att.: **Uzgeņerētu datu paraugs**

E1: Raksturo vienmērīgu, konstantu trafikas plūsmu, pievienojot augošo uzbrukumu, kas sākas laikā $\tau_1 = 0$ un beidzas laikā $\tau_2 \in (6000, 7000)$.

E2: Raksturo trafikas plūsmu ar augošo trendu,pievienojot augošo uzbrukumu, kas sākas laikā $\tau_1 = 0$ un beidzas laikā $\tau_2 \in (6000, 7000)$.

E3: Raksturo trafikas plūsmu ar dilstošo trendu, pievienojot augošo uzbrukumu, kas sākas laikā $\tau_1 = 0$ un beidzas laikā $\tau_2 \in (6000, 7000)$.

3.3. Datu transformācijas

Veicot tīkla trafika plūsmas analīzi, mēs esam spiesti strādāt ar ļoti lielu datu apjomu. Tas var stipri ietekmēt uz algoritma izpildes ātrumu. Tāpēc ir vērts samazināt datu apjomu ar kuru mēs veiksim analīzi, bet tajā pašā laikā nezaudējot informācijas par sākuma datu raksturu, dabu. Savā darbā es apskatīšu F^0,F^1-transformācijas un tas dažādas kombinācijas.

- Izmantojot, $F_9[f_i]$(F^0-transformācija, reprezentācijas algoritms ir aprakstīts punktā 2.2), kur f_i ir i-ta laikrinda. Tad ir iespējams reprezentēt datus no 10000 elementiem ar kortežu, kas sastāv no 9 elementiem. Dati pēc $F_9[f]$ transformācijas ir apskatāmi 3.3. attēlā.

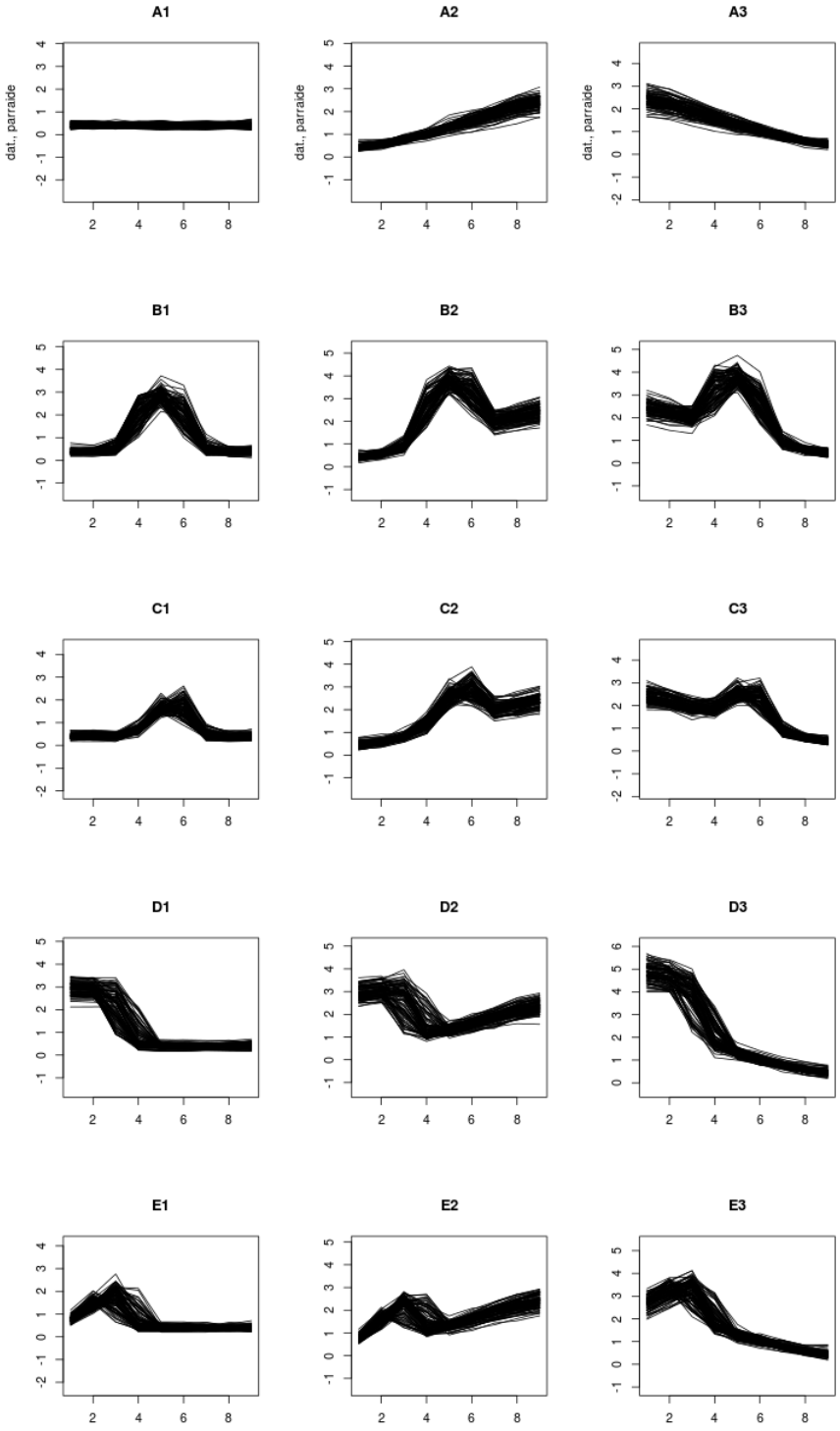
- Apskatīsim F^1-transformācijas komponentes, c^0, c^1, izvietojot tas secībā

$$[c_1^0, .., c_9^0, c_1^1, .., c_9^1].$$

  Citiem vārdiem tas ir vektors $c^0 \sharp c^1$, kas ir c^0 un c^1 konkatenācija. Mēģināsim novērtēt c^1 komponentes lomu analīzē. Tātad apskatīsim arī transformācijas ar tādu pašu uzbūvi, bet komponentes c^1 reizināsim ar 100 un 1000.

- Apskatīsim F^1-transformācijas tikai otru komponenti c^1, kas sniedz informāciju par datu izmaiņas ātrumu, jo c^1 raksturo atvasinājumu. Tātad mums būs pāreja uz kortēžu $c^1 = [c_1^1, .., c_9^1]$.

Tracformācijas programmas kods ir ievietots 1.pielikumā. F^0-transformētus datus ir iespējams apskatīt 3.3. attēlā, bet citus trasformācijas attēlus ir iespējams apskatīt 2. pielikumā.



3.3 att.: F_9^0**-transformētu datu paraugs**

4. TRANSFORMĒTU DATU KLASTERIZĀCIJAS ANALĪZE

4.1. Klasterizācija balstīta uz F-transformācijas koeficientu kombinācijām

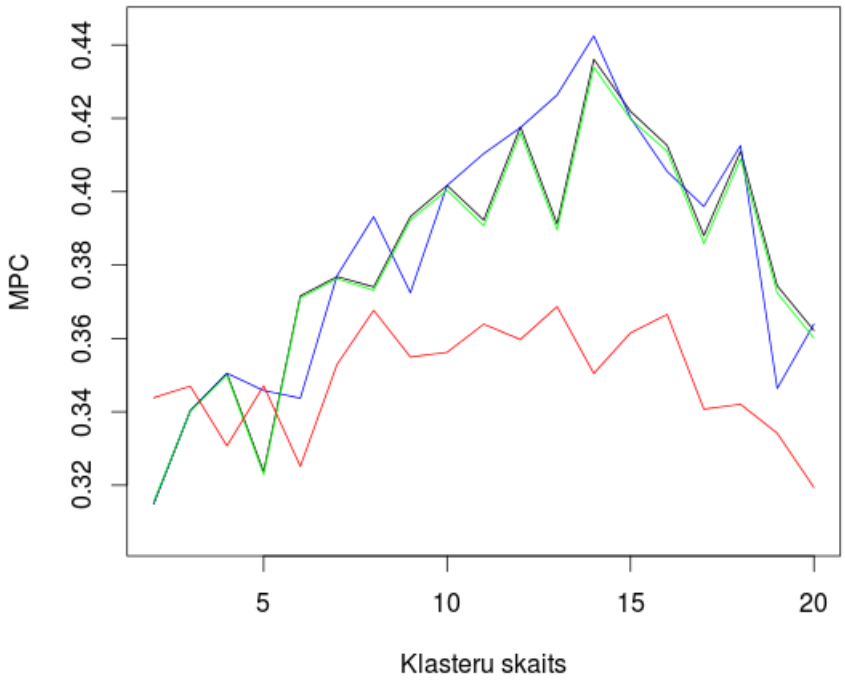
Trasformētu datu klasterizācijai izmantosim augstāk aprakstīto FCM algoritmu. Ieviesim apzīmējumus priekš izmantotām transformācijām:

I lietosim gadījumā, kad izmantosim F^0,

II lietosim gadījumā, kad izmantosim vektoru c^0 un c^1 konkatenāciju,

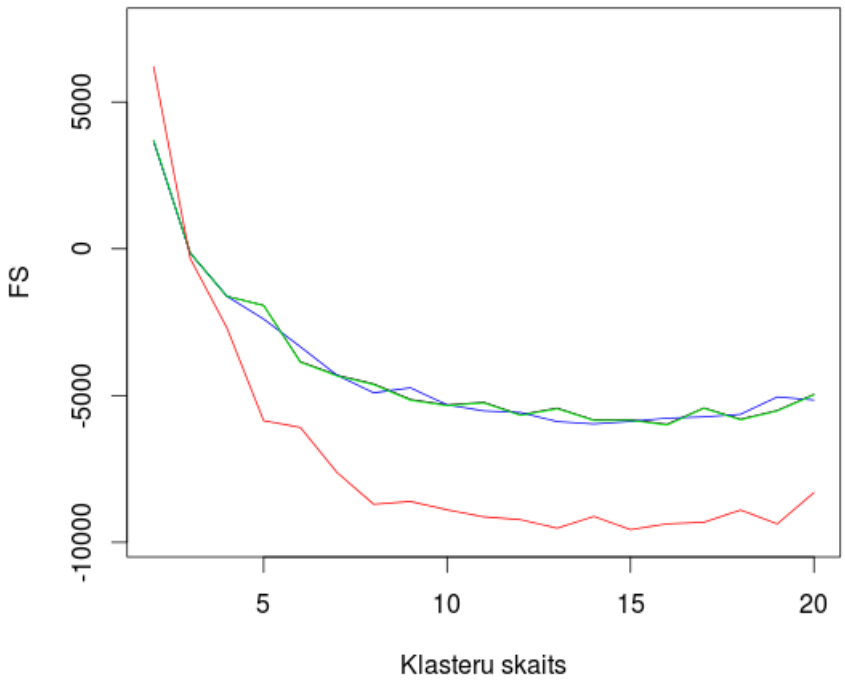
III lietosim gadījumā, kad izmantosim vektoru c^0 un c^1 konkatenāciju, vektors c^1 būs reizināts ar svaru 100,

IV lietosim gadījumā, kad izmantosim vektoru c^0 un c^1 konkatenāciju, vektors c^1 būs reizināts ar svaru 1000.



4.1 att.: **MPC validācijas indekss (I - melnā, II - zilā, III- zaļā, IV - sarkanā).**

Klasterizācijas algoritma darbībai un validācijas indeksu aprēķināšanai tika izmantots programmas kods, kas bija izmantots darbā [4]. Uz klasterizētiem datiem tika uztaisīta validācija pēc MPC, FS un XB indeksiem, 4.1 un 4.2. attēlos ir grafiski attēloti MPC,FS indeksi, bet 4.3. attēlā ir XB indekss.



4.2 att.: **FS validācijas indekss (I - melnā, II - zilā, III- zaļā, IV - sarkanā).**

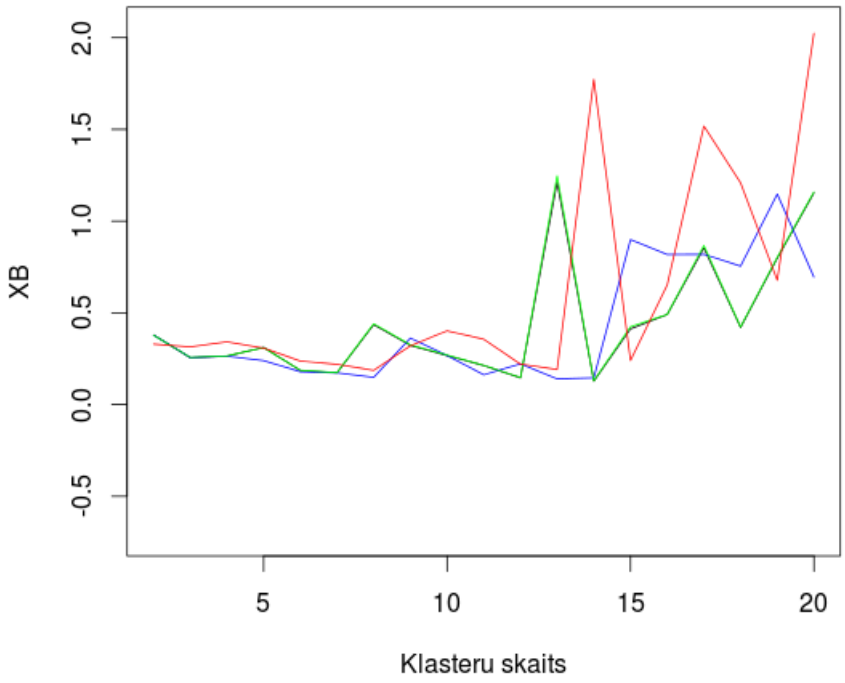
1 tabula: **Validācijas rezultāti.**

	MPC	**FS**	**XB**
I	14	16	14
II	14	14	13
III	14	16	14
IV	13	15	8

Validācijas rezultāti ir apkopoti 1. tabulā. No validācijas indeksu grafikiem un tabulas datiem, iespējams uztaisīt secinājumu, ka dati ir klasterizēti, kā bija sagaidāms, 15 klasteros. Bet, lai pārliecināties un parbaudīt klasteru sadalījumu, uzģenerēsim pārbaudes datus un uzstaisīsim klasifikāciju. Pārbaudes datu ģenerēšanas parametri atbilst pamatdatu ģenerēšanas parametriem.

Klasificējot pārbaudes datus izrādās, ka neskatoties uz to, ka pēc validācijas indeksiem ir atdalīti 15 klasteri, ir iespējami gadījumi, kad daži uzģenerēti dati neatdalās kā atsevišķa klase un tiek klasificēti citā, bet daži sašķēlās uz diviem. Tāpēc bija jāpārbauda visas transformācijas, mēģinot atrast to, kura dos labākus rezultātus.

Tabulās 2, 3 un tabulā 4 ir apkopota informācija par pārbaudes datu klasifikāciju, ņemot klasteru skaitu pēc MPC, FS un XB indeksiem attiecīgi. Salīdzinot tabulas, varam secināt, kā indeksi XB un MPC labāk atbilst patiesībai.



4.3 att.: **XB validācijas indekss (I - melnā, II - zilā, III- zaļā, IV - sarkanā).**

2 tabula: **Klasifikācijas rezultāti, kur klasteru skaits ir paņemts pēc MPC indeksa**

	I	II	III	IV
$A1$	atpazīts	atpazīts	atpazīts	atpazīts
$A2$	atpazīts	atpazīts	atpazīts	atpazīt
$A3$	atpazīts	atpazīts	atpazīts	atpazīts
$B1$	atpazīts	atpazīts	atpazīts	atpazīts
$B2$	atpazīts	atpazīts	atpazīts	atpazīts
$B3$	atpazīts	atpazīts	atpazīts	atpazīts
$C1$	atpazīts	atpazīts	atpazīts	atpazīts
$C2$	atpazīts	atpazīts	atpazīts	**A2 vai B2**
$C3$	atpazīts	atpazīts	atpazīts	**A2 vai B2**
$D1$	atpazīts	**A3 vai E2**	atpazīts	atpazīts
$D2$	**E2 vai E3**	atpazīts	**E2 vai E3**	atpazīts
$D3$	atpazīts	atpazīts	atpazīts	atpazīts
$E1$	atpazīts	atpazīts	atpazīts	atpazīts
$E2$	atpazīts	atpazīts	atpazīts	atpazīts
$E3$	atpazīts	atpazīts	atpazīts	atpazīts

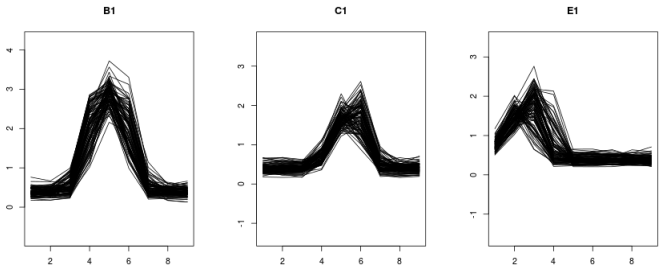
3 tabula: **Klasifikācijas rezultāti, kur klasteru skaits ir paņemts pēc FS indeksa**

	I	II	III	IV
$A1$	atpazīts	atpazīts	atpazīts	atpazīts
$A2$	atpazīts	atpazīts	atpazīts	atpazīt
$A3$	atpazīts	atpazīts	atpazīts	atpazīts
$B1$	**C1**	atpazīts	atpazīts	atpazīts
$B2$	atpazīts	atpazīts	atpazīts	atpazīts
$B3$	atpazīts	atpazīts	atpazīts	atpazīts
$C1$	atpazīts	atpazīts	**B1**	atpazīts
$C2$	atpazīts	atpazīts	atpazīts	**A2 vai B2**
$C3$	atpazīts	atpazīts	atpazīts	**A2 vai B2**
$D1$	atpazīts	**A3 vai E2**	**sadalās divos**	atpazīts
$D2$	atpazīts	atpazīts	atpazīts	**sadalās divos**
$D3$	atpazīts	atpazīts	**sadalās divos**	atpazīts
$E1$	atpazīts	atpazīts	atpazīts	atpazīts
$E2$	**sadalās divos**	atpazīts	atpazīts	atpazīts
$E3$	**sadalās divos**	atpazīts	atpazīts	**sadalās divos**

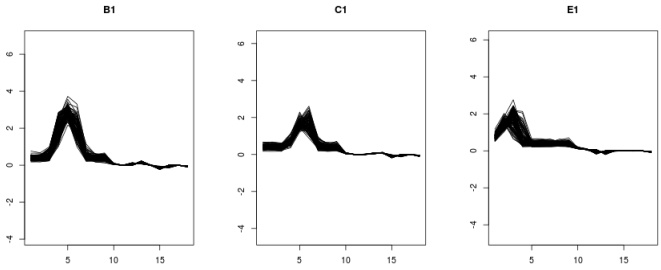
4 tabula: **Klasifikācijas rezultāti, kur klasteru skaits ir paņemts pēc XB indeksa**

	I	II	III	IV
$A1$	atpazīts	**apvienots ar E1**	atpazīts	**apvienots ar E1**
$A2$	atpazīts	atpazīts	atpazīts	**apvienots ar E2**
$A3$	atpazīts	atpazīts	atpazīts	**apvienots ar D1, E3**
$B1$	atpazīts	atpazīts	atpazīts	**apvienots ar C1**
$B2$	atpazīts	atpazīts	atpazīts	**apvienots ar C2**
$B3$	atpazīts	atpazīts	atpazīts	**apvienots ar C3**
$C1$	atpazīts	atpazīts	atpazīts	**apvienots ar B1**
$C2$	atpazīts	atpazīts	atpazīts	**apvienots ar B2**
$C3$	atpazīts	**A3 vai B3**	atpazīts	**apvienots ar B3**
$D1$	atpazīts	atpazīts	atpazīts	**apvienots ar A3, E3**
$D2$	**E2 vai E3**	atpazīts	**E2 vai E3**	atpazīts
$D3$	atpazīts	atpazīts	atpazīts	atpazīts
$E1$	atpazīts	**apvienots ar A1**	atpazīts	**apvienots ar A1**
$E2$	atpazīts	atpazīts	atpazīts	**apvienots ar E2**
$E3$	atpazīts	atpazīts	atpazīts	**apvienots ar A3, D1**

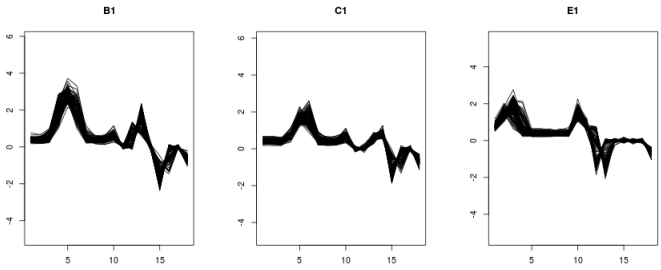
Spriežot pēc tabulas datiem pie visām apskatītām transformācijam ”labi” trafiki tika atpazīti kā atsevišķa klase. Tomēr iespējams izdalīt transformācijas ar labākiem rezultātiem. Tas ir I un III, kas dod līdzīgus rezultātus. Tā, piemēram, pie klasteru skaita 14, $F_9[f]$-transformētiem datiem neidentificējas kā atsevišķa klase tikai D1. FCM algoritms uzskatā, ka D1 attiecās pie E3 vai E2, kas arī ir ”uzbrukuma” klases. Turpretīm gadījumā II pie 14 klasteriem D1 varētu būt atpazīts kā A3 vai B3. Tas nozīmē, ka ir iespēja uzbrukumam būt piešķirtam pie normāla trafika paraugiem. Tātad, spriežot pēc iegūtiem rezultātiem, varam secināt, ka uz klasterizācijas kvalitāti vairāk ietekmē F^1-transformācijas pirmā komponente, tas ir c^0. Tomēr otrā komponente arī var būt derīga, ja mēs veiksim salikto klasterizāciju, pirmajā posmā mēģinot atrast izmaiņas laikrindu uzvedībā. Bet otrajā posmā jau pielietojot F-transformāciju.



(a) **ar I transformēti dati**



(b) **ar III transformēti dati**



(c) **ar IV transformēti dati**

4.4 att.: **Dažādas datu transformācijas**

5 tabula: **Klasifikācijas rezultāti**

(a) c^1 **klasifikācijas rezultāti.**

1	a_1,c_1,e_1
2	c_2
3	a_3,d_1,e_1,e_3
4	b_3
5	c_3
6	d_1,d_3
7	a_2,c_2,e_2
8	b_2
9	d_2,d_3,e_3
10	a_3,d_1,d_2,e_3
11	b_1

(b) c^1 **klasifikācijas rezultāti, neņemot pirmo un pedējo punktu.**

1	a_1,a_2,a_3
2	b_1,b_2,b_3
3	a_3,d_1,e_1,e_3
4	d_1,d_2,d_3
5	e_1,e_2,e_3
6	d_1,d_2,d_3,e_2,e_3

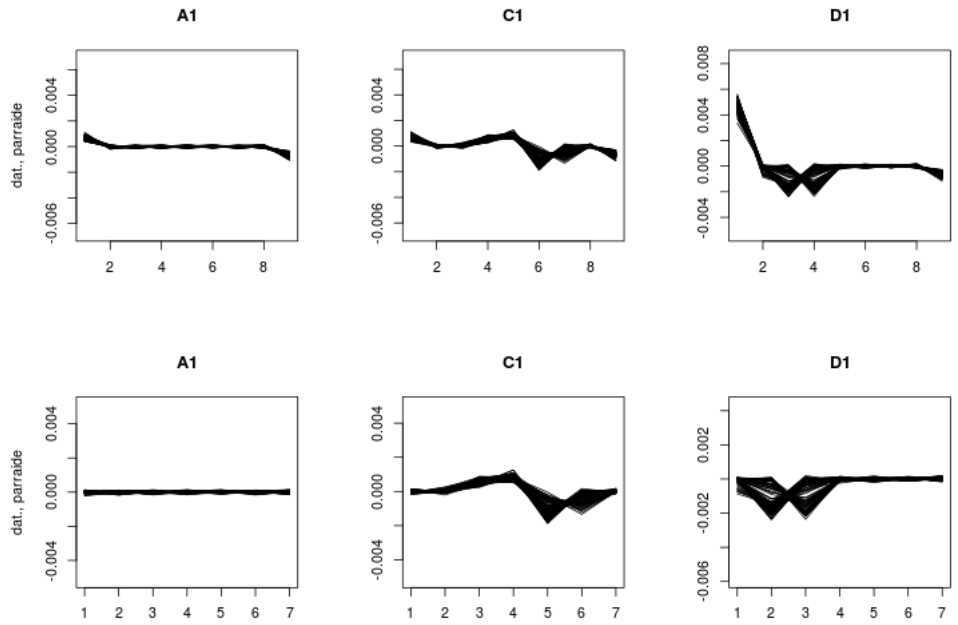
4.2. Klasterizācija balstīta uz F^1-transformācijas otro koeficientu

Pārbaudot c^1 transfomāciju, mēs sagaidām saņemt 5 klasterus, kas skaidros trafika uzvedību, tas ir A,B,C,D,E. Tāpat kā iepriekšējās transformācijas gadījumā izmantotas pārbaudes dati: $a_i, b_i, c_i, d_i, e_i, i = 1, 2, 3$. Tā kā pēc sākuma datu klasterizācijas validācijas indekss XB bija vienāds ar 11. Tad pārbaudes datu klasifikāciju veicām 11 klasteros.

Kā redzams no 5a. tabulas, izdalīti klasteri ir slikti atdalīti viens no otra, jo pārbaudes dati parasti pieder dažiem klasteriem vienlaicīgi. Un nekādā gadījumā mēs nevaram sagrupēt iegūtos 11 klasterus, lai tie veidotu 5 sagaidāmos, t.i., A,B,C,D,E.

Bet tomēr sanāk, ka mēs sasniegsim rezultātus, kas būs līdzīgi ar sagaidāmiem, ja paņemsim c^1 bez c_1^1 un c_n^1. Tas ir saistīts ar to, ka rēķinot c_i^1 koeficientus (sk., formulu (8)) mums ir svarīga informācija par to, kas notiek intervālā $[t_{i-1}, t_{i+1}]$, bet c_1^1 un c_n^1 gadījumā informācija nav pilna. Attēlā 4.5 ir redzamas dažas atšķirības starp transformētiem datiem. Ar c^1 transformētus datus ir iespējams apskatīt 2. pielikumā.

Kā redzams tabulā 5b, labota transformācija gandrīz labi uzdot pirmo tuvinājumu tālākai analīzei. Ja paskatīties kā ir klasificēti dati, tad ir redzams, ka tas ir klasteri A,B,C,D,E,E/D. To var izmantot kā pirmo priekšstatu. Tomēr, ja atgriezties pie iepriekšējām transformācijam un izlabot c^1, neņemot pirmo un pēdējo elementu, tad tas īpaši neuzlabos situāciju, kas vēlreiz liecina, ka datu analīzei vairāk svarīgākās ir c^0 komponentes.



4.5 att.: **Atšķirība c^1 transformācijā ar c_1^1, c_n^1 elementiem un bez tiem.**

4.3. Secinājumi

- Izmantojot transformācijas, kas pārveido sākuma datus c^0 un c^1 komponentes, pirmā komponente vairāk ietekmē uz klasterizācijas un klasifikācijas rezultātiem.
- Veicot klasteranalīzi ar c^1 palīdzību ir noderīgi neizmantot c_1^1 un c_n^1, pirmo un pēdējo elementu.
- F^1-transformācijas c^1 komponente var būt noderīga daudzkārtā klasterizācijā. Pirmajam priekšstatam izmantot c^1 un detalizētākai analīzei jau izmantot F^0-transformāciju.

NOBEIGUMS

Darbā tika parbaudītas F-transformācijas, pielietojot tas FCM klasterizācijas algoritmam. Bet pastāv vēl citi nestrikti klasterizācijas algoritmi, piemēram, iespējamību c-vidējo algoritms, modificētais iespējamību klasterizācijas algoritms un uzlabotais kombinētais nestriktais iespējamību algoritms, kas varētu dot labākus rezultātus. Kā arī var mēģināt uzlabot rezultātus mainot metriku. Šajā darbā bija izmantota Eiklīda metrika, bet mainot to uz citu iespējams uzlabot rezultātus. Tā, piemēram, Eiklīda metrikas vietā iespējams paņemt Mahalanobisa metriku, tāda klasterizācijas algoritmu modifikācija ir zināma kā Gustafsona-Kessela modifikācija.

Literatūras saraksts

[1] A. Šostaks, L-kopas un L-vērtīgas struktūras. Rīga: Latvijas Universitāte, 2003.

[2] I. Perfilieva, V. Novak, V. Pavliska, A. Dvorak, M. Štepnička, Forecasting Time Series Using Fuzzy Transform.
http://www.neural-forecasting-competition.com/downloads/NN3/methods/11 NN3 Perfilieva NN3.pdf [skatīts 01.03.2015]

[3] M. Štepnička, V. Pavliska, V. Novak, I. Perfilieva, L. Vavričkova, I. Tomanova, Time Series Analysis and Prediction Based on Fuzzy Rules and the Fuzzy Transform, Research report No. 139, 2009.
http://irafm.osu.cz/research report/141 rep139.pdf [skatīts 02.03.2015]

[4] J. Litviņenko, Nestriktas klasterizācijas iteratīvās metodes un to lietojumi DDoS uzbrukumu analīzē, Bakalaura darbs, Rīga: Latvijas Universitāte, 2014.

[5] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Kluwer Academic Publishers Norwell, 1981.

[6] W. W., Yunjie Zhang, On Fuzzy Cluster Validity Indices, Fuzzy Sets and Systems, 158, 2095-2117, 2007.

[7] J. M. Keller et al., A Fuzzy K-Nearest Neighbor Algorithm, IEEE Transactions on Systems, Man, and Cybernetics, 15(4), 580-585, 1985.

[8] I. Perfilieva, M. Dankova, Towards F-transform of a Higher Degree. Proceeding of the Congress IFSA-EUSFLAT, Lisbon, 585-588,2009.

[9] Milana Politehniskās universitātes mājas lapa.
http://home.deib.polimi.it/matteucc/Clustering/tutorial html/cmeans.html [skatīts 01.05.2015]

PIELIKUMI

1.Pielikums

Datu transformācju realizācijas R vidē

```
# distance if sigma=I - Euclidean, if sigma not= I - Mahalanobis
distance<-function(x,y,sigma) { # for Fuzzy Transform
  (t(x-y)%*%solve(sigma)%*%(x-y))
}

t<-8
n<<-10000
int<<-n/t


partition<-c()
partition[1]<-0

for (i in 1:(t+1)){
  partition[i+1]<-i*int
}

A.memb<-function(point,k,partition){
  A<-0
  h.k<-partition[k+1]-partition[k]

  if ((point>=partition[k])&(point<=partition[k+1])){
    A<-1-(point-partition[k])/h.k
  }else if (k==1){A<-0
  }else{
    if (((point>=partition[k-1])&(point<=partition[k]))&(k!=1)){
      h.previous<-partition[k]-partition[k-1]
      A<-(point-partition[k-1])/h.previous
    }}
 A
}

#F^0 transform

fTr<-list()
```

```
for ( i in 1:length(dat) ){
  fTr[[i]]<-c(1)

  for (k in 1:(t+1)){
    sUp<-0
    sLw<-0
    for (j in 1:n){
      sUp<-sUp+dat[[i]][[j]]*A.memb(j,k,partition)
      sLw<-sLw+A.memb(j,k,partition)
    }
    fTr[[i]]<-c(fTr[[i]],(sUp/sLw))
  }
  fTr[[i]]<-fTr[[i]][-1]
}
# End of F^0

# F^1=c0+c1(x-xk) transform c0 = F^0
f.Tr1<-list()
fTr1<-list()
for ( i in 1:length(dat) ){
  fTr1[[i]]<-c(1)


  for (k in 1:(t+1)){
    sUp<-0
    sLw<-0
    sUp1<-0
    sLw1<-0
    for (j in 1:n){

      sUp1<-sUp1+dat[[i]][[j]]*
        (j-partition[k])*A.memb(j,k,partition)
      sLw1<-sLw1+((j-partition[k])^2)*A.memb(j,k,partition)
    }
    fTr1[[i]]<-c(fTr1[[i]],(sUp1/sLw1))
  }
  fTr1[[i]]<-fTr1[[i]][-1]
}

mu<-1# mu=(1,100,1000)
```

```
################ F0 # F1

f.Tr1<-list()
f.Tr1<- fTr
for (i in 1:length(fTr1)){
  for (j in 2:(t)){
    f.Tr1[[i]][t+j]<-mu*fTr1[[i]][j]
  }
}
################ F0[i]+F1[i]


f.Tr1<-list()
f.Tr1<- fTr
for (i in 1:length(fTr)){
  for (j in 1:(t+1)){

    f.Tr1[[i]][j]<-fTr[[i]][j]+fTr1[[i]]
    [j]*((partition[j+1]-partition[j]))
  }
}

########################## F1[i] i=2,..,n-1
f1<-list()

for (i in 1:length(fTr1)){
  f1[[i]]<- fTr1[[i]][-1]
  f1[[i]]<- f1[[i]][-8]
}
################

X<-c()

for ( i in 1:length(fTr) ) X<-rbind(X,fTr[[i]])
for ( i in 1:length(fTr1) ) X<-rbind(X,fTr1[[i]])
for ( i in 1:length(f.Tr1) ) X<-rbind(X,f.Tr1[[i]])
for ( i in 1:length(f1) ) X<-rbind(X,f1[[i]])
```

Prototipiskās klasifikācijas realizācija R vidē

```
membership.prob<-function(X,C,sigma=sgm) {
  U<-matrix(nr=nrow(C),nc=nrow(X))
  for ( j in 1:nrow(X) ) {
    s<-0
    for ( i in 1:nrow(C) ) {
      s<-s+distance(X[j,],C[i,],sigma[[i]])^(-1/(m-1))
    }
    for ( i in 1:nrow(C) ) {
      if ( distance(X[j,],C[i,],sigma[[i]])==0 ) {
        U[i,j]<-1
      } else {
        U[i,j]<-((distance(X[j,],C[i,],sigma[[i]]))^(-1/(m-1)))/s
      }
    }
  }
  U
}

m<<-2
sgm<-list()
for ( q in 1:c ) {
  sgm[[q]]<-diag(l)
}
sgm<<-sgm
############################
RES<-list()
cc<-13
for (kk in 1:15){
  dat<-list()
  dat<-data[[kk]]

  #F^0 transform

  fTr<-list()
  for ( i in 1:length(dat) ){
    fTr[[i]]<-c(1)

    for (k in 1:(t+1)){
```

```
      sUp<-0
      sLw<-0
      for (j in 1:n){
        sUp<-sUp+dat[[i]][[j]]*A.memb(j,k,partition)
        sLw<-sLw+A.memb(j,k,partition)
      }
      fTr[[i]]<-c(fTr[[i]],(sUp/sLw))
    }
    fTr[[i]]<-fTr[[i]][-1]
  }
  ## End of F^0, instead F^0 might be any other transform !!!!!

  Y<-c()
  for ( i in 1:length(fTr) ) Y<-rbind(Y,fTr[[i]])

  class.FCM<-membership.prob(Y,Clist.FCM[[cc]])
  RES[[kk]]<-class.FCM
}

B<-list()
B<-RES

k<-length(B[[1]][1,])
n<-length(B)

res<-c()

RESULT<-list()

for (i in 1:n){
  for (j in 1:k){
    res[j]<-which.max(B[[i]][,j])
  }
  sk<-1
  for(jj in 2:length(res)){
    if (res[jj]==res[jj-1]){sk<-sk+1}
  }
    if (sk==k){RESULT[[i]][i]<-"found"}else{RESULT[i]<-"not found"}
}
```

2.Pielikums

I transformācijas pārbaudes datu D2 klasifikācija pie 14 klasteriem

D2	1	2	3	4	5
[1,]	0.04952005	0.06292327	0.04930666	0.05338956	0.05088313
[2,]	0.09322745	0.14446069	0.10412247	0.10429261	0.12333474
[3,]	0.1765426	0.14643425	0.13302929	0.15785381	0.13504438
[4,]	0.03750115	0.04228161	0.03201148	0.0403058	0.03053376
[5,]	0.08553425	0.09018592	0.08848683	0.08836494	0.09005169
[6,]	0.07804247	0.05401985	0.04980212	0.07237655	0.04284064
[7,]	0.03948182	0.03736145	0.03372316	0.04007265	0.03099654
[8,]	0.07980394	0.08539546	0.14630259	0.08107099	0.16870549
[9,]	0.04691525	0.03486879	0.08908895	0.04520302	0.06866868
[10,]	0.05349995	0.03337021	0.03963816	0.04776743	0.03217724
[11,]	0.05258548	0.04011652	0.05711217	0.04985499	0.04913003
[12,]	0.07160665	0.10100124	0.08557316	0.08181577	0.0921647
[13,]	0.09315971	0.08165601	0.05703557	0.09251451	0.05243923
[14,]	0.04257925	0.04592473	0.03476739	0.04511737	0.03302975

II transformācijas pārbaudes datu C3 klasifikācija pie 13 klasteriem

	1	2	3	4	5
[1,]	0.0374513	0.02984173	0.02757992	0.03648502	0.0303422
[2,]	0.29679353	0.20712797	0.1463793	0.33259748	0.48357846
[3,]	0.06881199	0.05513331	0.05370448	0.05970789	0.04399018
[4,]	0.03812884	0.03688407	0.04564154	0.04353796	0.0278418
[5,]	0.02747284	0.03382302	0.0436794	0.02990494	0.02171696
[6,]	0.04175706	0.05349594	0.0782025	0.04801149	0.032612
[7,]	0.15369608	0.23727432	0.2319169	0.12479955	0.10796269
[8,]	0.03273614	0.03475285	0.04261362	0.03492453	0.02393441
[9,]	0.04892595	0.04705145	0.05379808	0.04751237	0.03293187
[10,]	0.05220244	0.06796407	0.08415931	0.06123692	0.04689784
[11,]	0.03362644	0.0308669	0.03445816	0.03928923	0.0270955
[12,]	0.05671793	0.06540593	0.07024045	0.05286759	0.04204275
[13,]	0.11167948	0.10037844	0.08762633	0.08912503	0.07905332

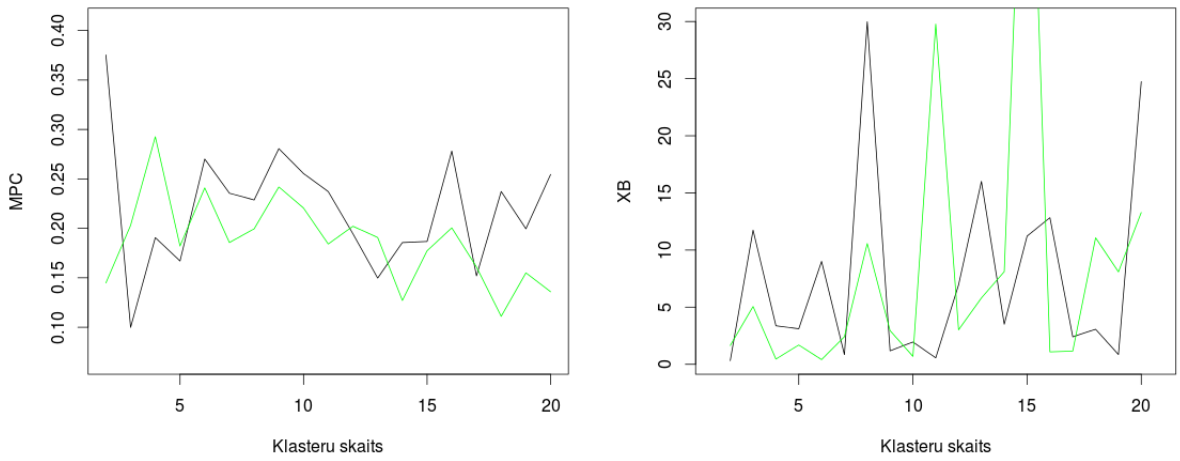
II transformācijas pārbaudes datu A1 un E1 klasifikācija pie 13 klasteriem

A1	1	2	3	4	5
[1,]	0.0002453488	0.0004853933	0.0004856077	0.0009226891	0.000289917
[2,]	0.000433087	0.0008384709	0.0008716681	0.0016228058	0.0005006051
[3,]	0.0004532781	0.0008743792	0.0009012349	0.0017170132	0.0005282285
[4,]	0.0006152164	0.0011721074	0.0012485718	0.0024124362	0.0006817452
[5,]	0.9878828842	0.9773703221	0.9751995315	0.9543143572	0.9865181834
[6,]	0.0041723461	0.0074253457	0.0088265454	0.0156821504	0.0043158421
[7,]	0.0011536737	0.0021965538	0.0023079928	0.0041660618	0.0013843235
[8,]	0.0011298075	0.002122178	0.0022937077	0.0044912894	0.0012466622
[9,]	0.000790505	0.0014983356	0.0015965551	0.0030480027	0.0009053556
[10,]	0.0012008643	0.0022805667	0.0024686171	0.0045122156	0.0013408811
[11,]	0.000383223	0.0007405567	0.0007750792	0.0014890293	0.000430369
[12,]	0.0009724851	0.0018904873	0.0018955862	0.0035262404	0.0011794841
[13,]	0.0005672807	0.0011053034	0.0011293024	0.0020957089	0.0006784034

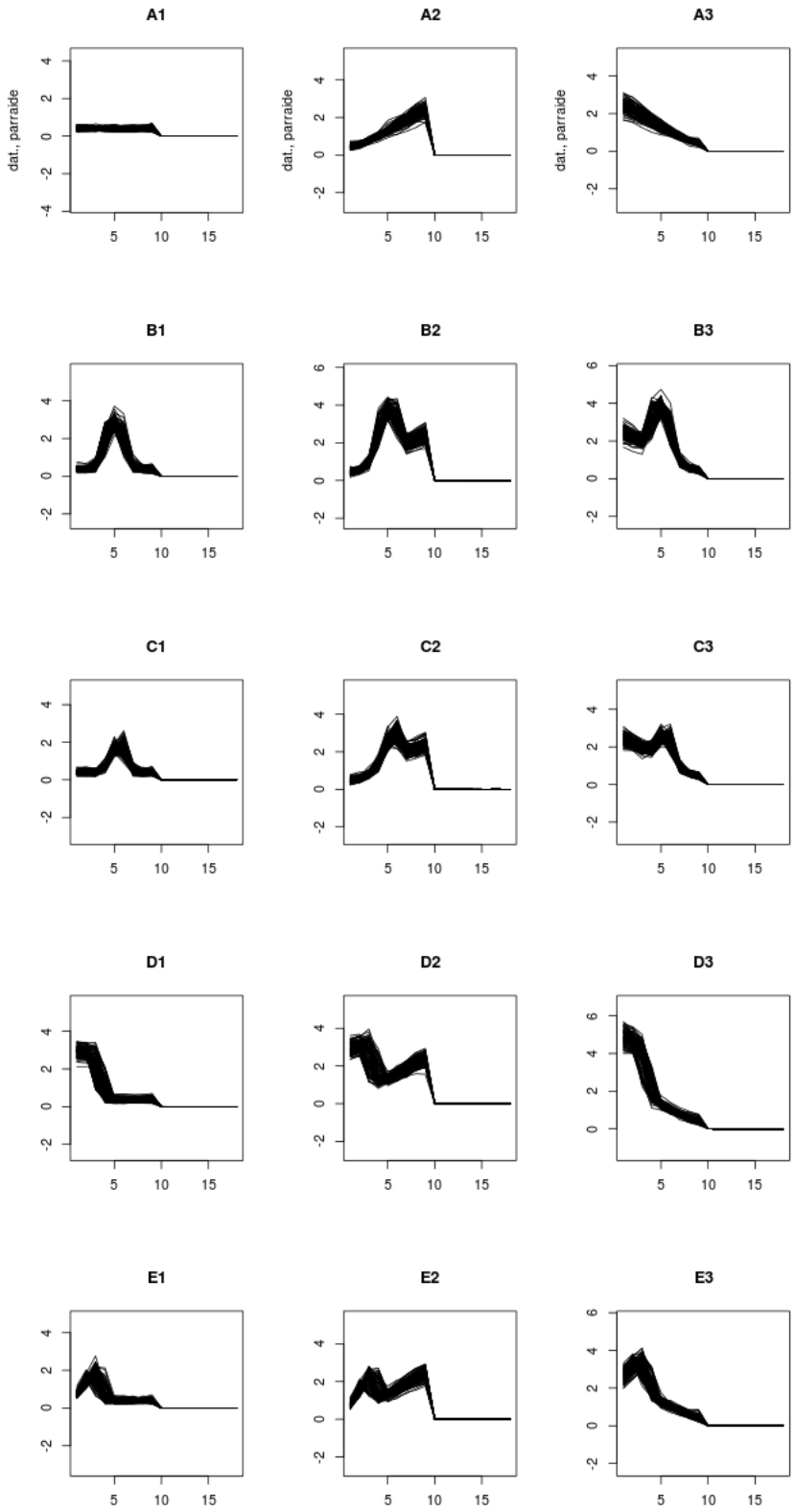
E1	1	2	3	4	5
[1,]	0.02117509	0.02035347	0.02260377	0.02041427	0.01900466
[2,]	0.03318223	0.03009352	0.0307033	0.03002502	0.02777453
[3,]	0.03464332	0.03803578	0.03825052	0.03567431	0.03183612
[4,]	0.03270938	0.03060667	0.03077614	0.031121	0.02915145
[5,]	0.29617478	0.24913261	0.26855683	0.30063539	0.35647698
[6,]	0.12311731	0.11556414	0.11038675	0.11760949	0.11998456
[7,]	0.11047043	0.13923856	0.12256983	0.11479379	0.09787508
[8,]	0.05370264	0.05218392	0.05109544	0.05285435	0.04855476
[9,]	0.05583404	0.05434093	0.05565246	0.05643331	0.0492461
[10,]	0.06814761	0.05266297	0.05327071	0.05667124	0.05434876
[11,]	0.02294165	0.019805	0.02054839	0.02074713	0.01949932
[12,]	0.08847925	0.13763752	0.1302831	0.10371283	0.09424843
[13,]	0.05942227	0.06034492	0.06530277	0.05930787	0.05199925

II transformācijas pārbaudes datu D1 klasifikācija pie D1 14 klasteriem

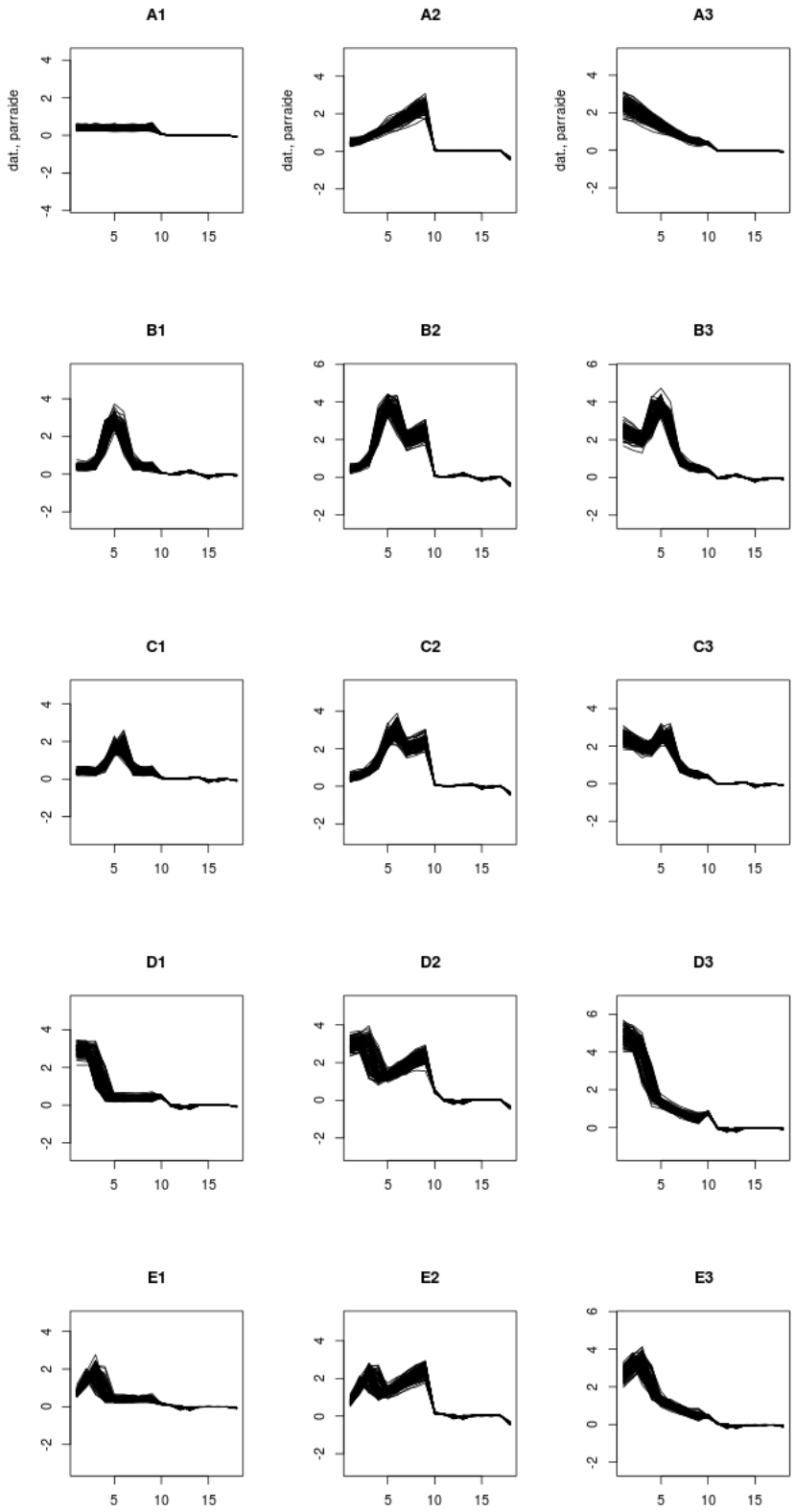
[1,]	0.01989351	0.02670313	0.03465029	0.03843767	0.03003864
[2,]	0.37408055	0.24134091	0.11345641	0.11331122	0.17936572
[3,]	0.0251612	0.02774157	0.02715066	0.03088345	0.02690436
[4,]	0.17881675	0.24745102	0.24329718	0.23405697	0.2411461
[5,]	0.02448163	0.03610381	0.05549898	0.06010483	0.04309664
[6,]	0.06372723	0.05333103	0.03405711	0.03869995	0.0423937
[7,]	0.01711228	0.01960668	0.0236961	0.0261739	0.02064323
[8,]	0.05840977	0.10467399	0.18449056	0.15507657	0.16462154
[9,]	0.0557181	0.06281553	0.0629431	0.07014647	0.05950906
[10,]	0.02278488	0.02683873	0.03522811	0.03779073	0.02951369
[11,]	0.08387467	0.05838858	0.04984528	0.05352007	0.05099974
[12,]	0.03608549	0.0366112	0.04090384	0.0421978	0.0387582
[13,]	0.02693768	0.04380706	0.07818421	0.08101983	0.05788652
[14,]	0.01291626	0.01458674	0.01659816	0.01858053	0.01512286



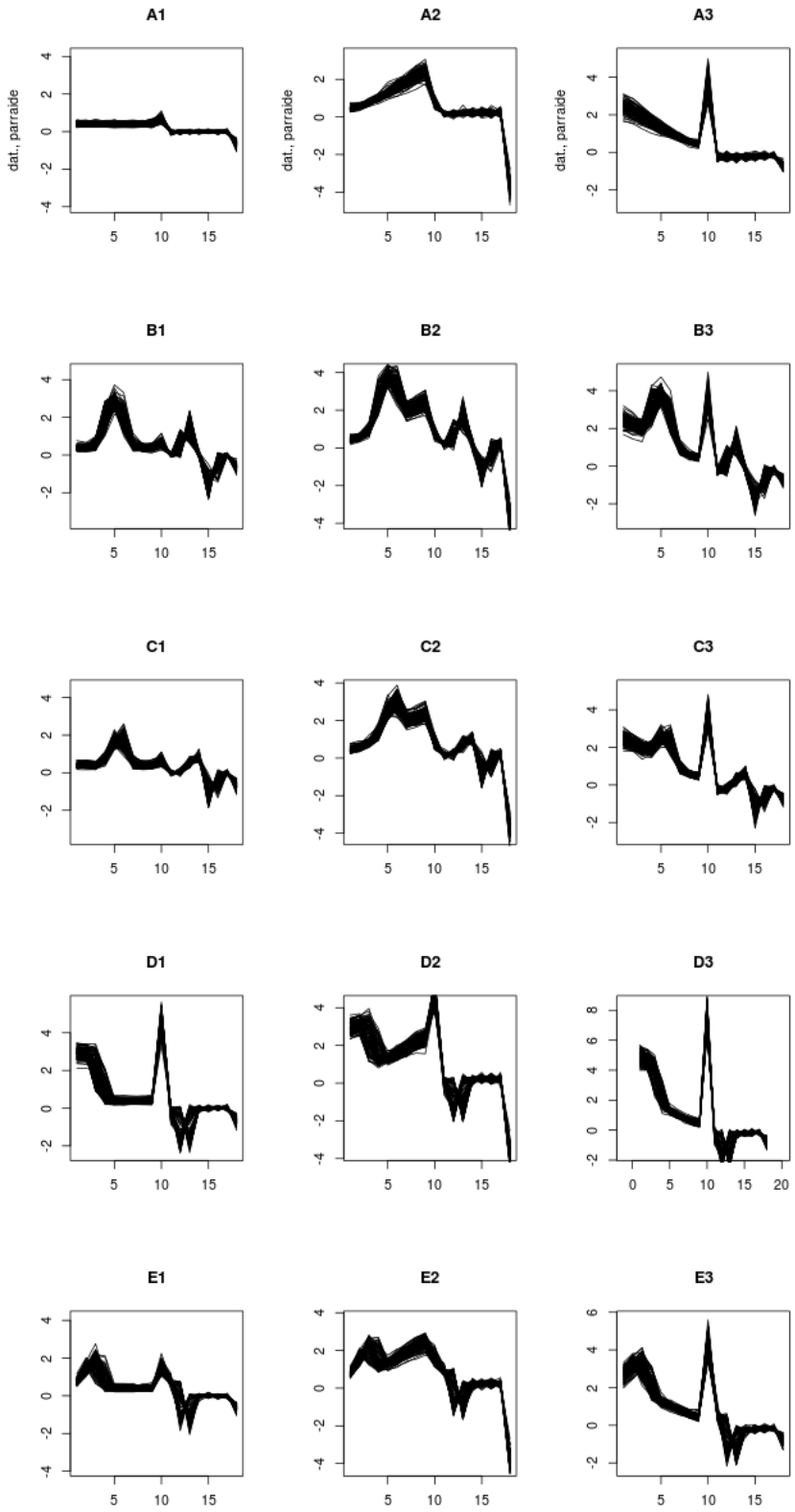
MPC un XB validācijas indeksi (izmantojot c^1 - melnā, c^1 izņemot c_1^1, c_n^1 - zaļā



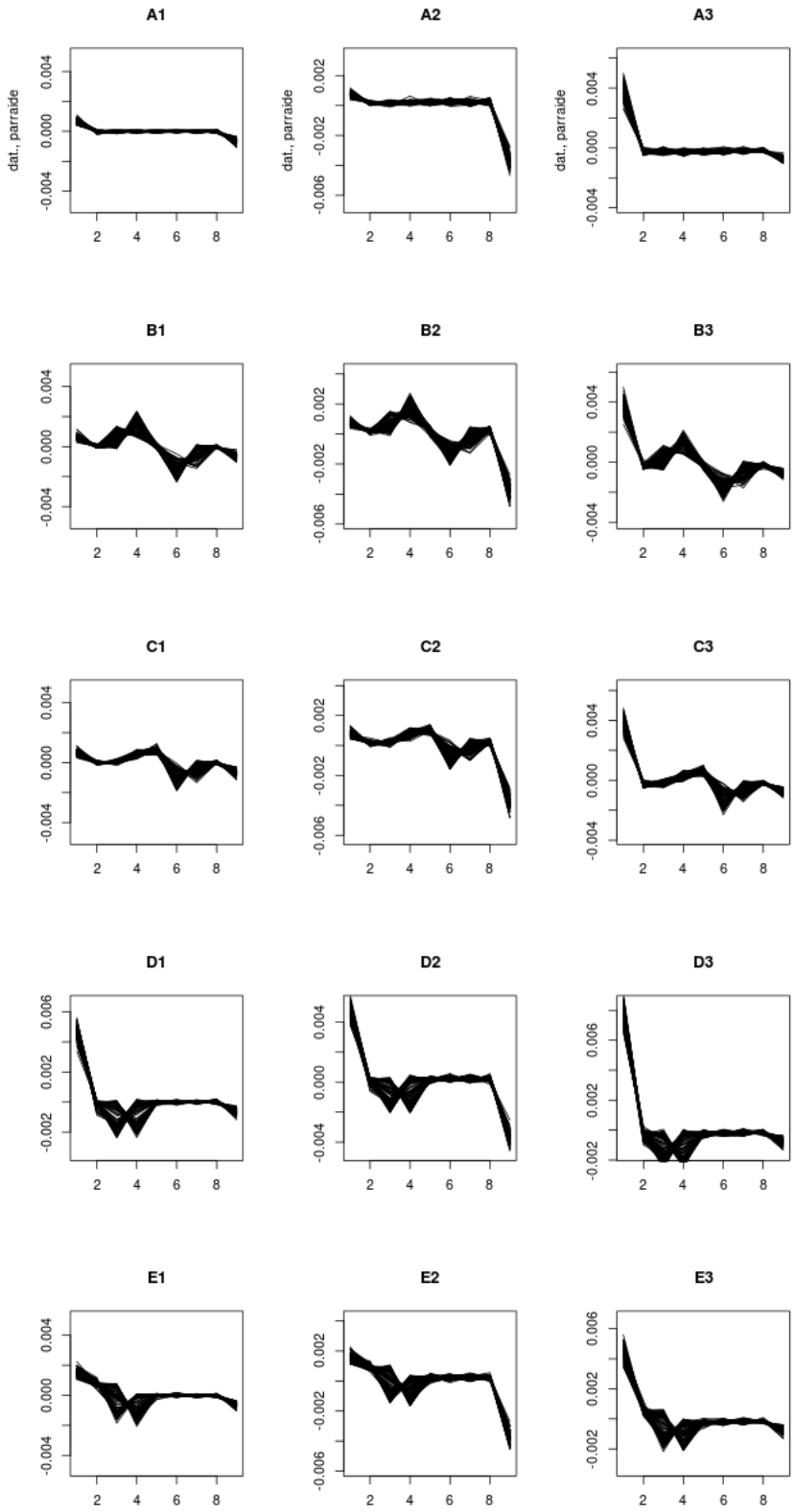
Transformācijas II iedarbības rezultāti.



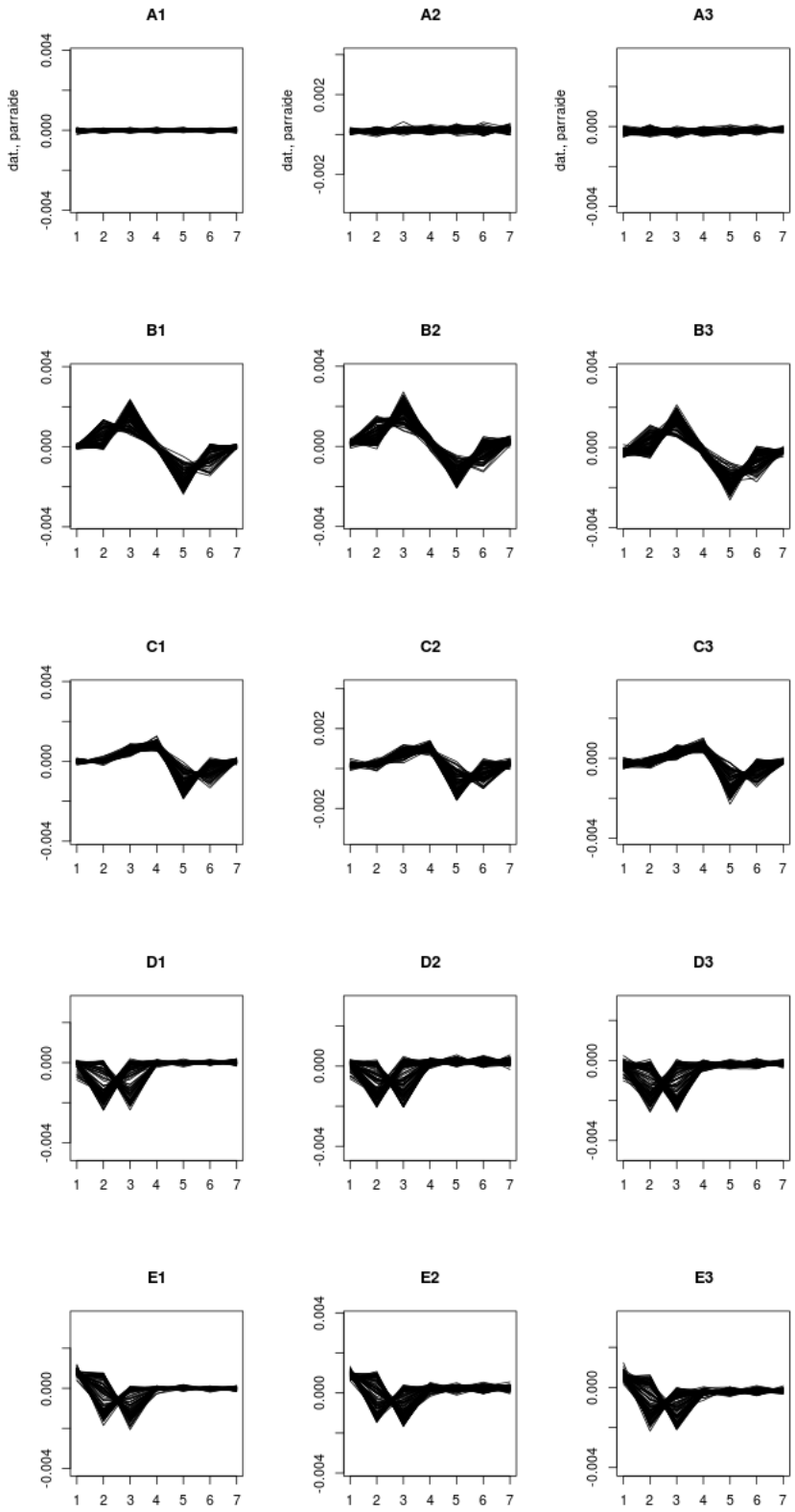
Transformācijas III iedarbības rezultāti.



Transformācijas IV iedarbības rezultāti.



Dati, kas ir reprezentēti ar c^1 vektoru.



Dati, kas ir reprezentēti ar c^1 vektoru, neņemot c_1^1, c_n^1 komponentes.

Bakalaura darbs ”Nestriktas klasterizācijas metodes balstītas uz F-transformētiem datiem” izstrādāts LU Fizikas un matemātikas fakultātē.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantojot tikai tajā norādītie informācijas avoti un iesniegta elektroniskā kopija atbilst izdrukai.
Autors _____ Mihails Anufrijevs

Rekomendēju darbu aizstāvēšanai
Vadītāja: profesore, Dr. Math. Svetlana Asmuss _____

Recenzente: docente, Dr. Math. Ingrīda Uļjane

Darbs iesniegts Matemātikas nodaļā __.06.2015.
Dekāna pilnvarovā persona: vecākā metodiķe Dzintra Holsta

Darbs aizstāvēts bakalaura gala pārbaudījuma komisijas sēdē
12.06.2015. prot. Nr._____
Komisijas sekretāre: docente Margarita Buiķe