



SÄÄTÖTEKNIIKAN LABORATORIO

Neuraalilaskenta ja epälineaarinen
dynamiikka
komponenttien kulutus- ja
myyntiennusteiden
laatimisessa

Petri Mutka

Raportti B No 19, Joulukuu 2000

Oulun yliopisto
Säätötekniikan laboratorio
Raportti B No 19, Joulukuu 2000

Neuraalilaskenta ja epälineaarinen dynamiikka komponenttien kulutus- ja myyntiennusteiden laatimisessa

Petri Mutka
Säätötekniikan laboratorio, Oulun yliopisto,
PL 4300, 90014 Oulun yliopisto

Tiivistelmä: Elektroniikkateollisuuden yksi perusongelma on komponenttien pitkä hankinta-aika. Lyhyt toimitusaika johtaa väistämättä vaihto-omaisuuden kasvuun, ellei komponenttien hankinta-aikaa pyritä lyhentämään. Komponenttien hankinnan ja varastoinnin hallintaan tarvitaan ennakoivia menetelmiä. Tämän tutkimuksen tavoitteena oli selvittää komponenttien kulutushistoriatietojen käyttöön soveltuvia ennustamismenetelmiä. Laskenta-ajan lyhentämiseksi komponentit pyrittiin jakamaan luokkiin niiden ennustettavuuden perusteella. Sopiville komponenteille pyrittiin laatimaan ennusteet käytön kannalta sopivassa aikaikkunassa.

Tutkimuksessa tarkasteltiin tilastollisia menetelmiä ja erilaisia neuroverkkoja. Heuristisesti suoritettu luokittelu ei ollut riittävän tarkka ja yksiselitteinen, joten luokittelun tarkentamiseksi pyrittiin luomaan opetus- ja testiaineisto eksaktien parametrien mukaan. Tässä raportissa esitellään useita analysointimenetelmiä ja osoitetaan testitapauksilla, että niiden avulla voidaan erottaa ennustettavissa oleva aika-sarja sellaisista, joita ei voi ennustaa eksaktisti. Testitapauksina käytettiin tunnettuja kaootisia kuvauksia ja yhtälöryhmiä. Eri menetelmien yhdistelmänä kehitettiin aikasarjan kompleksisuuden analysointiin soveltuva yleistetty menetelmä.

Varsinaisen sovelluksen kannalta ongelmana on selkeästi datan saatavuus. Niille komponenteille, joista dataa oli edes likimain tarpeeksi, saatiin ennusteajoiksi vain muutamia päiviä, joka oli reilusti alle vaaditun tavoitteen. Suurimmasta osasta komponentteja ei voitu sanoa mitään saatavilla olevan datan perusteella. Raportti sisältää listan niistä komponenteista, jotka osoittivat jonkun asteista sisäistä korrelaatiota. Dataa komponenttien kulutushistoriasta ei ollut saatavilla tarpeeksi pitkältä ajalta. Datan kompleksisuus on myös niin korkea, että vaadittujen ennusteajojen saavuttaminen on mahdotonta.

Tulokset perustuvat Oulun yliopiston ”Työkaluympäristöt ja kehitysmenetelmät” (TOOLMET) projektin, joka kuului osana TEKES:n teknologiaohjelmaan ”Oppivien ja älykkäiden järjestelmien sovellukset”. Projektin rahoituksesta tämän sovelluksen osalta vastasivat TEKES ja Nokia Access Systems Oy,

Hakusanat: neuraalilaskenta, epälineaarinen dynamiikka, elektroniikkateollisuus.

ISBN 951-42-5873-8
ISSN 1238-9404
ISBN 951-42-7502-0 (PDF)

Oulun yliopisto
Säätötekniikan laboratorio
Prosessi- ja ympäristötekniikan osasto
PL 4300
90014 Oulun yliopisto

SISÄLLYSLUETTELO

1	Johdanto	4
2	Projektin kulku	5
3	Käytetty data komponenttikannasta	7
4	Neuroverkot	10
4.1	Luokittelu	10
4.2	Ennakointi	11
5	Analysointimenetelmät	12
5.1	Lineaarinen autokorrelaatiofunktio	12
5.2	Average Mutual Information	12
5.3	Shannon-informaatio	13
5.4	Algorithmic Complexity	13
5.5	False Nearest Neighbors (FNN)	14
6	Testitapaukset	16
6.1	Lorenz-yhtälöt	16
6.2	Hénon-kartta	20
6.3	Ikeda-kartta	23
6.4	Kohina	27
6.5	Esimerkkitapausten tulkinta	30
7	Komponenttikannan analysointi	33
7.1	Analyysi	33
7.2	Analysoinnin tulokset	36
8	Yhteenveto	44
	Lähdeluettelo	46
	Liite 1. Komponenttilista	47

1 Johdanto

Elektroniikkateollisuuden yksi perusongelma on komponenttien pitkä hankinta-aika. Asiakastyytyväisyyden ylläpitämiseksi tavoiteltu lyhyt toimitusaika johtaa väistämättä vaihto-omaisuuden kasvuun, ellei komponenttien hankinta-aikaa pyritä lyhentämään. Yksi mahdollisuus on kehittää komponenttien hankinnan ja varastoinnin hallintaa toimivaksi *ennakoivien menetelmien* avulla.

Täsmällisesti määriteltynä ennustaminen tai ennakoiminen tarkoittaa seuraavaa: joidenkin tämänhetkisten tai menneisyydestä saatujen tietojen avulla pyritään arvioimaan jonkun systeemin tulevaa käyttäytymistä tietyllä aikavälillä. Tämä tarkoittaa sitä, että ennustamisessa tarvittava tieto *on jo olemassa*. Periaatteessa aikasarja (komponentin kulutushistoria, valuuttakurssien vaihtelut tms.) heijastelee sitä informaatiota, joka sen on luonut. Eli voimme valita minkä tahansa pisteen aikasarjalta, ja päättää, että tästä ajanhetkestä aikaisemmat pisteet ovat historiaa ja myöhäisemmät "tulevaisuutta". Näin voimme etsiä mahdollisia kytkentöjä tulevaisuuden ja menneisyyden välillä, eli mekanisme, joka aiheuttaa aikasarjan sisäisen (*auto*)*korrelaation*.

Jos aikasarjan menneisyys ei korreloi tulevaisuuden kanssa (eli peräkkäisten datapisteen välinen korrelaatio on nolla), on kyseessä puhdas *kohina*. Kohinasta voidaan sanoa vain se, että se on täysin satunnainen otanta arvoja joltain väliltä, joka noudattaa jotain jakaumaa ja muita tilastollisia tunnusmerkkejä. Sen eksaktia aikariippuvaa käyttäytymistä ei voi ennustaa. Toisessa ääripäässä on aikasarja, jonka kaikki pisteet korreloivat hyvin vahvasti toisiensa kanssa (esimerkiksi ajan mukana vakiona pysyvä funktio).

Näistä esimerkistä lähtöisin olevia välimuotoja voidaan tarkastella analysoimalla sen mekanismin *monimutkaisuutta*, joka aikasarjan aiheuttaa. Käytännössä voidaan puhua *kaaoksen asteesta* tai *kompleksisuudesta*. Matalan kaaoksen aikasarja voi silmämääräisesti näyttää satunnaiselta, mutta on olemassa analysointimenetelmiä, jotka kykenevät erottamaan matalan kaaoksen aikasarjan satunnaisesta aikasarjasta mikäli dataa on riittävästi. Lisäksi neuroverkko kykenee oppimaan tällaisten matalan kaaoksen aikasarjojen ennustamiseen jollain aikavälillä. Kaaoksen asteen kasvaessa lähestytään täysin satunnaista aikasarjaa, jonka käyttäytymistä ei ole mahdollista ennustaa eksaktisti. Käytännössä on olemassa seuraava nyrkkisääntö: mitä korkeampi kaoottisuus, sitä enemmän dataa tarvitaan aikasarjan käsittelyyn ja analysointiin.

Neuroverkkoprojektin perusajatuksena oli tutkia mahdollisuutta kehittää menetelmä käytettävien komponenttien määrien ennakoimiseen niiden kulutushistoriatietojen perusteella. Laskenta-ajan lyhentämiseksi komponentit pyrittiin jakamaan luokkiin niiden ennustettavuuden perusteella. Tämän jälkeen sopiville komponenteille laaditaan ennusteet. Projektin loppupuolella oli tarkoitus laatia menetelmä ennusteiden laatimiseksi. Vaadittu ennuste aika oli kaksi kuukautta (42 päivää).

Tämän varsin vaativan tehtävän ratkaisuvaiheet on esitetty seuraavassa kappaleessa. Laajojen menetelmävertailujen jälkeen päädyttiin ratkaisukokonaisuuteen, joka esitetään tässä raportissa.

2 Projektin kulku

Projekti aloitettiin tutkimalla ongelman käsittelyyn sopivaa kirjallisuutta sekä kokoamalla sovelluskelpoisia menetelmiä koskevia tietoja. Samalla pyrittiin hankkimaan mahdollisimman paljon dataa komponenteista menetelmien kehittelyä varten. Seuraavaksi hahmoteltiin pääpiirteittäin menetelmä, joka suorittaa ennusteiden laatimisen. Tässä vaiheessa päädyttiin jakoon *luokittelu - ennusteiden laatiminen*.

Ongelman käsittelyä jatkettiin tutkimalla erilaisia luokittelumenetelmiä. Ensimmäisenä kokeiltiin pienessä mittakaavassa aikasarjoista laskettujen *tilasollisten tunnuslukujen* (esim. keskiarvo, keskihajonta tms.) avulla muodostettuun koko aikasarjaa kuvaavaan tunnuslukuun perustuvaa luokittelua. Tämä menetelmä ei kyennyt luokittelemaan komponentteja halutulla tavalla.

Projektiä jatkettiin tarkastelemalla erilaisia *neuroverkkoihin perustuvia* luokittelumenetelmiä. Suuri osa tässä vaiheessa käytetystä ajasta kului aikasarjoista laskettavien tunnuslukujen kehittelyyn, joita käytettiin neuroverkolle syötettävänä parametreinä. Lisäksi verkkoja varten tarvittiin valmiiksi luokiteltu opetus/testiaineisto. Tässä vaiheessa testiaineisto laadittiin heuristisesti aikasarjan ominaisuuksia arvioimalla. Parhaimmillaan tämän kaltainen luokittelu saatiin toimimaan noin 90 prosentin tarkkuudella. Koska saavutettu tarkkuus ei ollut riittävä, päätettiin tarkastella uudelleen luokittelukriteerejä sekä parametreja. Luokitteluparametrien etsimisen ja testaamisen jälkeen päädyttiin siihen, että heuristisesti suoritettu luokittelu ei ollut riittävän tarkka ja yksiselitteinen.

Luokittelun tarkentamiseksi pyrittiin luomaan opetus- ja testiaineisto eksaktien parametrien mukaan: testi/opetusaineisto pyrittiin laatimaan valmiiden komponenttikohdistaisten *ennusteiden ominaisuuksien* perusteella. Tätä varten tutkittiin erilaisia neuroverkkopohjaisia ennustemenetelmiä. Ennusteiden laatimisessa esiintyneiden vaikeuksien takia aikasarjoja analysoitiin myös informaatioteoreettisten menetelmien avulla ja epälineaarista dynamiikkaa hyväksi käyttäen. Nämä menetelmät antoivat teoreettiset rajat tarvittavien datapisteiden määrälle, sekä mahdollisten ennusteiden pituuksille.

Tässä vaiheessa kärjistyi kriittiseksi projektia koko ajan vaivannut ongelma; saatavissa olevan *datan määrä*. Niille komponenteille, joista saatavilla olevaa dataa oli edes likimain tarpeeksi, saatiin ennustejoiaksi vain muutamia päiviä, joka oli reilusti alle vaaditun tavoitteen. Suurimmasta osasta komponentteja ei voitu sanoa mitään saatavilla olevan datan perusteella. Näinollen projektin loppuvaiheessa pyrittiin keskittymään komponenttikannan analysointiin, sekä niiden komponenttien seulomiseen, joiden kulutus *osoittaa edes jossain määrin säännönmukaisuutta*.

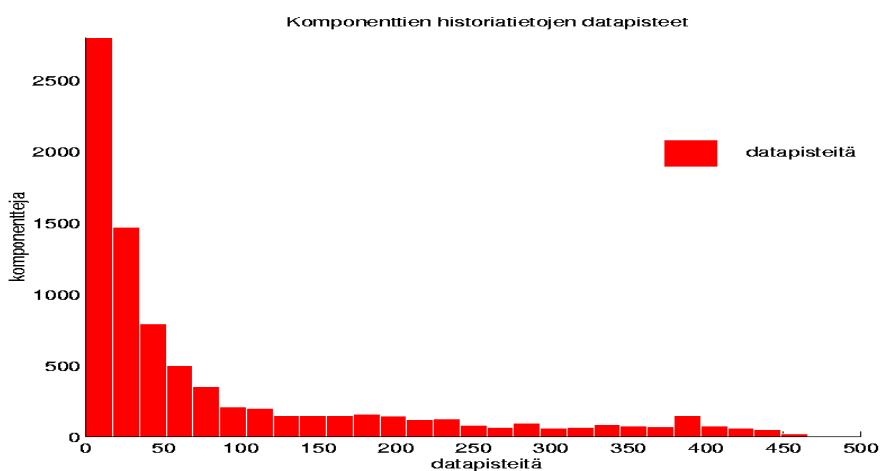
Projektissa tehty laskenta, ohjelmistokehittäminen ja datankäsittely tehtiin PC-tietokoneilla, joiden prosessoreiden (Intelin Pentium-prosessori) kellotaajuudet olivat 133MHz, 166MHz ja 200MHz. Kaikissa tietokoneissa oli käyttöjärjestelmänä Microsoftin Windows NT 3.51. Laskennassa, datan käsittelyssä, analysoinnissa ja visualisoinnissa käytettiin MathWorks Inc. valmistamaa Matlab 4.2 ohjelmistoa. Raskeampi laskenta suoritettiin C++ kielellä kirjoitetuilla ohjelmilla. Ohjelmien tekemiseen käytettiin Microsoftin Visual C++ 4.0 kääntäjää. Kaikki projektiin liittyvät dokumentit on kirjoitettu Microsoftin Word for Windows95 7.0 tekstinkäsittelyohjel-

malla. Projektiin liittyvässä visualisoinnissa on käytetty myös Microsoftin PowerPoint for Windows95 7.0 ohjelmaa.

Raportin rakenne on seuraava: ensin komponenttikannasta saatua dataa tarkastellaan tilastollisesti. Seuraavaksi tarkastellaan lyhyesti neuroverkkojen käyttömahdollisuuksia. Tutkimuksen keskeisissä kappaleissa esitellään käytetyt analysointimenetelmät ja osoitetaan testitapauksilla, että niiden avulla voidaan erottaa ennustettavissa oleva aikasarja sellaisista, joita ei voi ennustaa eksaktisti. Sen jälkeen esitetään komponenttikannasta tehty analyysi ja sen tulokset. Lopuksi kootaan tulokset ja analysoidaan niitä lyhyesti. Lisäksi tämä dokumentti sisältää listan niistä komponenteista, jotka osoittivat jonkun asteista sisäistä korrelaatiota.

3 Käytetty data komponenttikannasta

Komponenttikannan analysoinnissa käytetty data on haettu tietokannoista joulukuussa 1997. Haku on suoritettu siten, että jokaisesta komponentista on tiedostossa päiväkohtainen kulutus niin pitkältä ajalta kuin dataa on ollut saatavilla. Haun tuloksena saatiin komponentteja kaikkiaan 8322 kappaletta. Kaikki ne komponentit, joista on dataa vähemmän kuin 100 pistettä jätettiin kappaleessa 7 suoritetun analysoinnin ulkopuolelle. Perustelut valitulle rajalle on esitetty kappaleen 6 testitapausten perusteella.

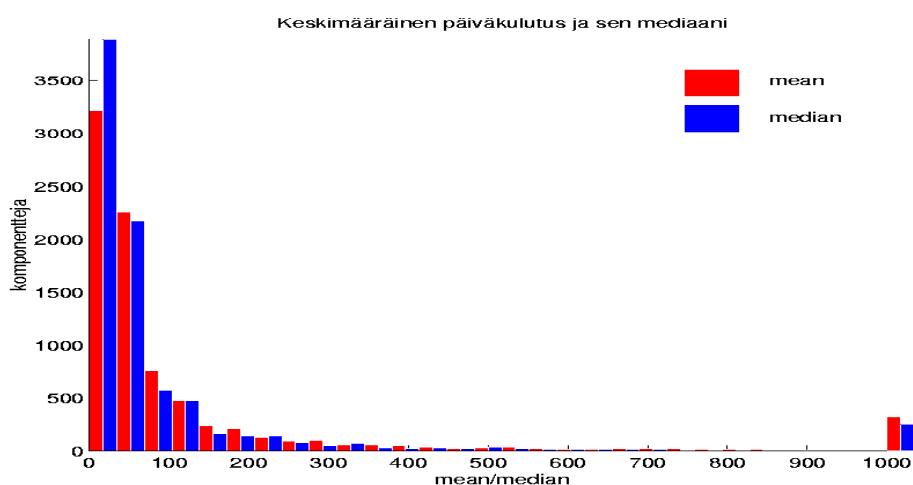


Kuva 1. Histogrammi komponenttien historiatietojen datapisteiden lukumäärästä.

Kuvassa 1 on histogrammi komponenttien kulutushistoriatietojen datapisteiden lukumäärien jakaumasta. Vaaka-akselilla on komponentin historiatiedoissa olevien datapisteiden lukumäärä, ja pystyakselilla on vastaavien komponenttien lukumäärä koko komponenttikannasta tehdyssä haussa. Samat tiedot on esitetty taulukossa 1. Suurin osa saatavilla olevasta datasta (noin 74 %) on alle edellä mainitun kriittisen rajan (taulukko 1). Analysoinnin piiriin jää 2135 komponenttia.

Taulukko 1. Datapisteiden määrä komponenttikannan historiatiedoissa.

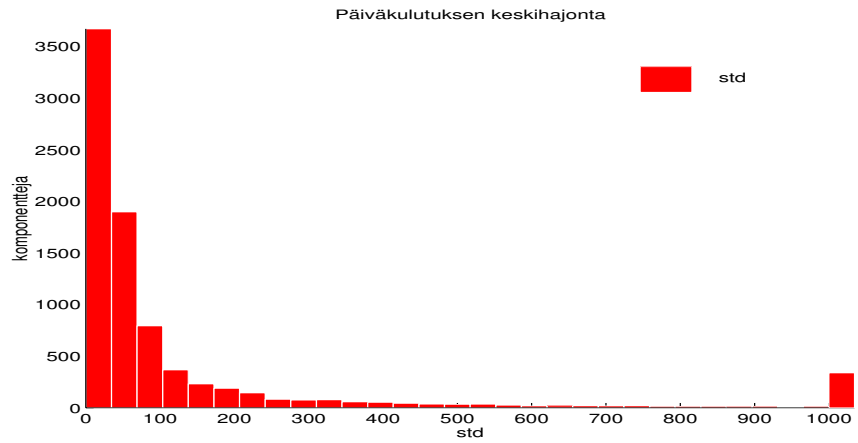
Datapisteitä	Komponentteja välillä	Komponentteja alle ylärajan (kumulatiivinen)	Komponentteja yli alarajan (kumulatiivinen)
0-50	5268 (63.30 %)	5268 (63.30 %)	8322 (100.00 %)
50-100	919 (11.04 %)	6187 (74.35 %)	3054 (36.70 %)
100-150	483 (5.80 %)	6670 (80.15 %)	2135 (25.65 %)
150-200	449 (5.40 %)	7119 (85.54 %)	1652 (19.85 %)
200-250	315 (3.79 %)	7434 (89.33 %)	1203 (14.46 %)
250-300	224 (2.69 %)	7658 (92.02 %)	888 (10.67 %)
300-350	208 (2.50 %)	7866 (94.52 %)	664 (7.98 %)
350-400	301 (3.62 %)	8167 (98.14 %)	456 (5.47 %)
400-450	151 (1.81 %)	8318 (99.95 %)	155 (1.86 %)
450-	4 (0.05 %)	8322 (100.00 %)	4 (0.05 %)



Kuva 2. Histogrammi komponenttien historiadatan keskiarvoista ja mediaaneista.

Luonnehditaan vielä datan laatua kahdella muulla kuvaajalla. Kuvassa 2 on histogrammi komponenttien historiadatan keskiarvojen ja mediaanien jakaumasta. Eli kyseessä on keskimääräinen päiväkulutus tai sen mediaani. Vaaka-akselilla on mediaani/keskiarvo ja pystyakselilla vastaavien komponenttien lukumäärä koko komponenttikannassa tehdyssä haussa.

Vertaamalla mediaania ja keskiarvoa toisiinsa saa jonkinlaisen kuvan datan hajonnasta. Kuvassa 3 on histogrammi komponenttien historiadatan, eli päiväkohtaisen kulutuksen, todellisista keskihajonnoista. Vaaka-akselilla on keskihajonta ja pystyakselilla vastaavien komponenttien lukumäärä koko komponenttikannassa tehdyssä haussa.



Kuva 3. Histogrammi komponenttien historiadatan keskihajonnoista.

4 Neuroverkot

Projektin aikana kokeiltiin useita erilaisia neuroverkkoihin perustuvia menetelmiä historiadatan luokitteluun sekä ennustamiseen. Kuten jo edellä tuli ilmi, menetelmät eivät kyenneet luokittelemaan tai ennustamaan aikasarjoja vaaditulla tavalla. Tämän takia testatut neuroverkkoihin perustuneet menetelmät esitellään vain lyhyesti tässä kappaleessa.

4.1 Luokittelu

Luokittelun ajatuksena oli nopeuttaa historiadatan käsittelyprosessia karsimalla etukäteen sellaiset komponentit, joiden kulutusta ei kyetä ennakoimaan. Tätä varten testattiin useita neuroverkkoihin perustuvia luokittelumenetelmiä.

Neuroverkko on oppiva järjestelmä; sen kokemus jonkin asian suorittamisessa varas-toituu *neuroneiden* välisien kytkentöjen painokertoimien arvoihin. Tämän *kokemuksen* avulla verkko suorittaa tehtävänsä. Komponenttien luokittelua varten tarvittiin etukäteen luokiteltu aineisto, jolla neuroverkkoa opetettiin. Tämä aineisto luotiin manuaalisesti käyttäen heuristisia luokitteluperusteita.

Neuroverkko vaatii numeerisessa muodossa olevaa tietoa käsiteltävästä kohteesta. Suuri osa testausta edeltävästä työstä meni sellaisten suureiden etsimiseen, joilla aikasarjan haluttuja ominaisuuksia voitaisiin kuvata. Luokittelua kehiteltäessä neuroverkoille syötettävänä tietoina yritettiin käyttää jotain yhdistelmää seuraavista parametreista:

- datapisteiden lukumäärä
- aikasarjan muodostaman kuvaajan ja aika-akselin rajoittaman pinta-alan suhde sellaisen suorakaiteen pinta-alaan, jonka korkeus on sama kuin aikasarjan maksimin arvo ja leveys aikasarjan pituus aikayksikköinä.
- aikasarjan muodostaman kuvaajan pituus suhde aikasarjan pituuteen aika-akselilla.
- aikasarjan keskihajonta normitettuna aikasarjan maksimin suhteen.
- aikasarjan keskiarvo normitettuna aikasarjan maksimin suhteen.
- aikasarjan mediaani normitettuna aikasarjan maksimin suhteen.
- Lombin periodigrammin viiden korkeimman piikin taajudet ja intensiteetit.
- *embedding schemen* mukainen tila-avaruuden dimensio d .
- korrelaatioidimensio (laskettu *embedding schemen* avulla).
- box-counting -dimensio.
- Capacity-dimensio (laskettu *embedding schemen* avulla).
- globaali Lyapunovin eksponentti.
- Kolmogorovin entropia (laskettu *embedding schemen* avulla).
- esikäsitellyn aikasarjan aikakomponentin keskiarvo (katso datan esikäsitteily kappaleesta 7 komponenttikannan analyysi).
- esikäsitellyn aikasarjan aikakomponentin mediaani (kappale 7).
- esikäsitellyn aikasarjan aikakomponentin keskihajonta (kappale 7).

Tässä vaiheessa täytyy huomauttaa, että *embedding schemestä* saadut arvot ovat vähintäänkin kyseenalaisia datapisteiden vähäisen määrän takia. Lisäksi *embedding scheme* menetelmän tekemien lineaaristen sovitusten automatisointi toimivaksi kaikissa tapauksissa oli erittäin vaikeaa. Parhaita luokitteluparametreja testausten perusteella näytti olevan datapisteiden lukumäärä, korrelaatioidimensio(!) sekä esikäsitellyn aikasarjan aikakomponenteistä lasketut suureet.

Luokittelun suorittamiseen kokeiltiin SOM (Self-Organizing Map) verkkoja ja backpropagation verkkoja useilla erilaisilla geometrioilla (verkon koko ja yksittäisten neuroneiden kytkentöjen lukumäärä) ja opetusalgoritmeilla;

- SOM-verkot klassisella opetusalgoritmillä, eri geometrioilla.
- SOM-verkot LVQ, LVQ2, LVQ2.1 ja LVQ3 algoritmeilla eri geometrioilla.
- Backpropagation verkot eri geometrioilla ja opetusalgoritmeilla (momentum, adaptive learning parameters).

Mikään näistä ei kyennyt luokittelua suorittamaan tarpeeksi tarkasti. Parhaimmat tulokset saatiin opettamalla SOM-verkkoa ensin LVQ3.1 algoritmillä ja sen jälkeen klassisella SOM-algoritmillä.

4.2 Ennakointi

Myös aikasarjojen ennustamisessa yritettiin käyttää neuroverkkoja. Tähän tarkoitukseen sovellettiin backpropagation ja FIR-backpropagation verkkoja (Wan 1990). FIR verkkojen perusajatuksena on korvata neuroneiden väliset kytkennät FIR-suodattimilla, joiden painot riippuvat neuroneiden edellisistä aktivoinneista. FIR-verkko on käytännössä sama asia kuin time-delay backpropagation verkko; erona on algoritmin esitys ja implementointi, joka on huomattavasti yksinkertaisempi FIR-verkkojen tapauksessa. Ennakoimisen perusajatuksena on antaa verkolle syöttöparametreina esimerkiksi viiden edellisen pisteen arvot, peräkkäisten arvojen erotukset tai jotain vastaavaa, ja opetetaan verkko niiden avulla "arvaamaan" seuraavan pisteen arvo.

Parhaiten ennakoimisessa menestyi FIR-verkko, joka on voittanut Santa Fe:n ennakointimenetelmien kilpailun. Menetelmä kykenee oppimaan Hénon-kartan rakenteen ja tuottamaan vastaavan attraktorikuvaajan (kuva 7). Mikään menetelmä ei kyennyt ennakoimaan testauksessa käytettyjen komponenttien kulutusta muutamaa pistettä pidemmälle (yksi datapiste on päivä).

5 Analysointimenetelmät

Tässä kappaleessa esitellään komponenttien historiadatan analysoinnissa käytetyt menetelmät. Näiden lisäksi datan käsittelyyn on kokeiltu muitakin menetelmiä, kuten Fourier-spektrin tai Lombin periodigrammien laskemista ilman menestystä.

5.1 Lineaarinen autokorrelaatiofunktio

Autokorrelaatio tarkoittaa aikasarjan tai funktion sisäistä korrelaatiota. Käytännössä se on aikaerotuksen funktio, joka kertoo sen kuinka paljon kaksi tietyn välimatkan päässä toisistaan olevaa pistettä korreloivat keskenään. Lineaarinen autokorrelaatiofunktio määritellään seuraavasti:

Merkitään dataa (aikasarjaa) symboleilla $s_i = [s_0, s_1, s_2, s_3, \dots, s_N]$. Viiva symbolin päällä merkitsee keskiarvoa. Lineaarinen autokorrelaatiofunktio $F(t)$ on siten

$$F(t) = \frac{\frac{1}{N+1} \sum_{i=0}^N (s_{i+t} - \bar{s})(s_i - \bar{s})}{\frac{1}{N+1} \sum_{i=0}^N (s_i - \bar{s})^2}. \quad (1)$$

Tässä \bar{s} on keskiarvo:

$$\bar{s} = \frac{1}{N+1} \sum_{i=0}^N s_i. \quad (2)$$

Voimakkaasti epälinearisilla funktioilla määritelty *lineaarinen autokorrelaatiofunktio* ei välttämättä kykene ilmaisemaan aikasarjassa olevaa informaatiota, koska funktion määritelmässä tehdään oletus lineaarisuudesta: kaksi s_i :n arvoa s_i ja s_{i+t} riippuvat toisistaan säännön $s_{i+t} - \bar{s} = F(t)(s_i - \bar{s})$ mukaisesti.

5.2 Average Mutual Information

Gallager esitti vuonna 1968 lineaarisen korrelaatiofunktion epälineaarisen yleistyksen (*Average Mutual Information*). Se mittaa systeemin tilan ennustettavuutta tai "muistia". Tätä varten systeemi täytyy jakaa tiettyihin diskreetteihin symboleihin. Kuvataan dataa (aikasarjaa) symbolijonolla $s_i = [s_0, s_1, s_2, s_3, \dots, s_N]$ ja merkitään kaikkia mahdollisia arvoja jotka s_i voi saada K :lla symbolilla w_i . Olkoot symbolin w_i esiintymistodennäköisyys symbolijonossa p_i ja symbolin w_j vastaava todennäköisyys p_j . Merkitään kytkettyä todennäköisyyttä sille, että symbolin w_i jälkeen n :n askeleen päässä esiintyy symboli w_j , merkinnällä $p_{ij}(n)$. Näinollen voidaan *Average Mutual Information* $I(t)$ määritellä seuraavasti:

$$I(t) = \sum_{i,j=0}^K p_{ij}(t) \log_2 \frac{p_{ij}(t)}{p_i p_j}. \quad (3)$$

Tämä toimii epälineaaristen systeemien kanssa huomattavasti lineaarista autokorrelaatiofunktioita paremmin, sillä kytketyt todennäköisyydet kuvaavat aikariippuvuuksia paremmin kuin oletus lineaarisesta kytkennästä (Schwarz et al. 1995, Abarbanel et al. 1993).

5.3 Shannon-informaatio

Jaetaan alkuperäinen aikasarja diskreetteihin symboleihin edellisessä kappaleessa kuvatulla tavalla. Tarkastellaan aikasarjassa esiintyviä n :n symbolin pituisia jonoja. Näinollen n :nnen asteen informaatio H_n määritellään seuraavasti:

$$H_n = - \sum_{s \in S_n} p_s \log_2 p_s . \quad (4)$$

Tässä S_n sisältää kaikki symbolijonot joiden pituus on n symbolia, ja p_s on kunkin symbolijonon esiintymistodennäköisyys aikasarjassa.

Shannon-informaatio H_n mittaa n merkkiä pitkän symbolijonon sisältämää keskimääräistä informaatiota. Informaatioiden H_n erotukset

$$h_n = H_{n+1} - H_n \quad (5)$$

konvergoivat n :n mukana systeemin Shannon-informaatioon h . Koska erilaisten symbolijonojen lukumäärä lisääntyy eksponentiaalisesti kasvatettaessa n :n arvoa, täytyy informaation arvioinnissa tyytyä muutaman ensimmäisen (N kappaletta) erotuksen h_n keskiarvoon:

$$\bar{h} = \frac{1}{N} \sum_{i=1}^N h_i = \frac{1}{N} \sum_{i=1}^N (H_{n+1} - H_n) \quad (6)$$

(Schwarz et al. 1995).

5.4 Algorithmic Complexity

Formaalisti symbolijonon kompleksisuus määritellään *sen lyhyimmän ohjelman pituutena bitteinä, joka kykenee tuottamaan ko. symbolijonon*. Lempel ja Ziv sovelsivat vuonna 1976 määritelmää käytäntöön käyttäen ohjelmaa joka osaa kaksi operaatiota: kopioi ja sijoita. Algoritmi toimii seuraavalla tavalla: Olkoot symbolijono $S = [s_0, s_1, s_2, s_3, \dots, s_N]$, joka jaetaan rekursiivisesti sanoihin w_α siten, että $S = [w_0, w_1, w_2, w_3, \dots, w_\mu]$. Sanat w_α ovat sellaisia, että $w_0 = s_0$, ja $w_{\alpha+1}$ on lyhyin mahdollinen sana, joka on w_α :n jälkeen ja jota ei voi muodostaa yhdistelemällä sanoista $w_0, w_1, w_2, w_3, \dots, w_\alpha$. *Algorithmic complexity* on näiden erilaisten sanojen lukumäärä $c(n)$, missä n on symbolijonon pituus (Kaspar ja Schuster 1987).

Lempel ja Ziv osoittivat analyttisesti, että satunnaiselle merkkijonolle:

$$\lim_{n \rightarrow \infty} c(n) = \frac{n}{\log_K n}. \quad (7)$$

Tässä K on käytössä olevien symbolien lukumäärä. Näinollen *algorithmic complexity* arvoa arvioitaessa erilaisten sanojen lukumäärän sijaan tarkastellaan suhteen c_r arvoa:

$$c_r = \frac{c(n) \log_K n}{n}. \quad (8)$$

Chaitin (1987) todisti, että Markov prosesseille (satunnaisprosessi, jonka seuraavan arvon todennäköisyys riippuu viimeisimmistä arvoista) *algorithmic complexity* ja *Shannon-informaatio* käyttäytyvät samalla tavalla.

5.5 False Nearest Neighbors (FNN)

Epälineaarisen dynamiikan perusajatuksena on aikasarjaa analysoitaessa siirtää datan kuvaus d -ulotteiseen avaruuteen (tila-avaruus). Kuvaus muodostaa radan ko. avaruudessa, jonka dynaamisia, globaaleja ja lokaaleja ominaisuuksia tutkimalla voidaan saada paljon informaatiota alkuperäisestä aikasarjasta. Transformaatiota, jossa aikasarja kuvataan d -ulotteiseen avaruuteen kutsutaan *embedding procedure*ksi. Varsinainen siirto on melko yksinkertainen, todellinen ongelma on siirron parametrien määrittäminen. Yksi näistä parametreista on tila-avaruuden dimensio d (Abarbanel et al. 1993, Hilborn 1994, Grassberger ja Procaccia 1983,1984, Eckmann ja Ruelle 1985, Farmer 1980).

Tila-avaruuteen siirtymistä hallitsee yksi keskeinen teoreema (*embedding theorem*):

Kaksi erillistä tila-avaruuden rataa eivät saa leikata toisiaan, eivätkä itseään, tila-avaruudessa.

Käytännössä yllä oleva teoreema tarkoittaa seuraavaa: jos tila-avaruuteen siirretyn aikasarjan tilakuvaaja leikkaa itsensä jollain ajan hetkellä, on tila-avaruuden dimensio liian pieni. Dimensio heijastelee niiden muuttujien lukumäärää, joita tarvitaan aikasarjan tuottaneessa systeemissä (esim. differentiaaliyhtälöryhmän muuttujien lukumäärä).

False nearest neighbors (FNN) on luotettava ja yksinkertainen menetelmä määrittää tarvittava tila-avaruuden dimensio. Se perustuu suoraan edellä esitettyyn teoreemaan (*embedding theorem*). Koska dimensio d kuvaa myös systeemin kompleksisuutta, voidaan menetelmää käyttää myös aikasarjan kompleksisuuden asteen arviointiin. Käytännössä menetelmässä tarkastellaan tila-avaruuden pisteitä ja tutkitaan ovatko ne läheisiä naapureita projektiossa käytetyn liian alhaisen dimension takia, ts. määritellään kriteeri sille, että pisteet ovat lähellä toisiaan projektion dimension takia, ja kas-

vatetaan dimensiota niin kauan, että tämän kriteerin täyttäviä pistepareja ei ole enää olemassa. Kennel et al. esitti vuonna 1992 algoritmin FNN-menetelmälle (Abarbanel et al. 1993).

Merkitään dataa (aikasarjaa) symboleilla $s_i = [s_0, s_1, s_2, s_3, \dots, s_N]$. Dimensiossa d kukin vektori y_i (tila-avaruuden piste) on muotoa:

$$y_i = [s_i, s_{i+\delta}, s_{i+2\delta}, \dots, s_{i+(d-1)\delta}] . \quad (9)$$

Tässä δ on muunnoksessa käytetty viive (*time delay*), joka on kokonaisluku. Jokaisella vektorilla y_i on lähin naapuri y_i^{NN} . Etäisyyksinä käytetään Euklidista etäisyyttä; näin on vektorin y_i ja sen lähimmän naapurin y_i^{NN} välinen etäisyys $R_i(d)$:lle voimassa:

$$R_i^2(d) = (s_i - s_i^{NN})^2 + (s_{i+\delta} - s_{i+\delta}^{NN})^2 + (s_{i+2\delta} - s_{i+2\delta}^{NN})^2 + \dots + (s_{i+(d-1)\delta} - s_{i+(d-1)\delta}^{NN})^2 . \quad (10)$$

Kun dimensiota kasvatetaan yhdellä, on etäisyys

$$R_i^2(d+1) = R_i^2(d) + (s_{i+d\delta} - s_{i+d\delta}^{NN})^2 . \quad (11)$$

Jos $R_i(d) \ll R_i(d+1)$, se johtuu siitä, että dimension kasvaessa on toisiaan liian lähellä olevat pisteet projektoitu kauemmas toisistaan. Kun valitaan tietty kynnyisarvo R_T etäisyyden muutokselle, saadaan ensimmäiseksi ehdoksi naapureiden “vääryydelle”:

$$\frac{|s_{i+d\delta} - s_{i+d\delta}^{NN}|}{R_i(d)} > R_T . \quad (12)$$

Käytännössä R_T :n arvo voidaan valita melkoisen vapaasti väliltä 10 - 50. Tämä kriteeri toimii jo yksin melkoisen hyvin, mutta sillä on yksi heikkous: esim. satunnaislukugeneraattorilla tuotettu korkeadimensioinen data tunnistetaan virheellisesti mataladimensioiseksi dataksi. Jos em. satunnaisdatan datapisteiden lukumäärää kasvatetaan, myös tuloksena saatu dimensio kasvaa. Tämä johtuu siitä, että, koska data on tasaisesti jakautunut (satunnaisdataa), jossain vaiheessa tulee tilanne, jossa kaikilla pisteillä on lähin naapuri, joka ei liiku kauemmaksi dimensiota kasvatettaessa. Jos dataa olisi ääretön määrä, ongelmaa ei olisi. Tarvitaan toinen ehto, eli pisteet ovat “väärää” naapureita, jos:

$$\frac{R_i(d+1)}{R_A} \geq 2 , \quad (13)$$

jossa

$$R_A^2 = \frac{1}{N+1} \sum_{i=0}^N (s_i - \bar{s})^2 . \quad (14)$$

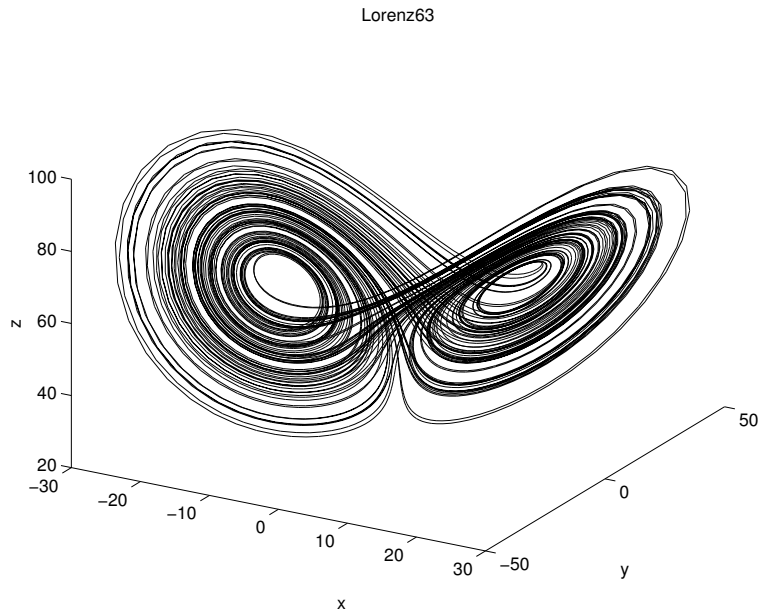
Tässä \bar{s} on yhtälön 2 mukainen keskiarvo. Lisäksi tarvitaan tietysti myös epäyhtälön 12 mukainen ehto (Abarbanel et al. 1993).

6 Testitapaukset

Tässä kappaleessa käsitellään muutama hyvin tunnettu testitapaus (kaoottinen kuvaus tai yhtälöryhmä tms.) kappaleessa 5 esitetyillä menetelmillä, ja osoitetaan se, että niillä voidaan erottaa matalan ja korkean kaaoksen aikasarjat toisistaan. Käsiteltävät tapaukset ovat *Lorenz*-malli (kappale 6.1), *Hénon*-kartta (kappale 6.2), *Ikeda*-kartta (kappale 6.3) ja tasaisesti jakautuneista satunnaisluvuista muodostettu kohina (kappale 6.4). Esimerkkitapauksien yhteenveto on kappaleessa 6.5.

6.1 Lorenz-yhtälöt

Vuonna 1963 MIT:n tutkija Edward Lorenz esitteli yksinkertaisen mallin, jolla mallinnettiin ilmakehän käyttäytymistä ja siinä tapahtuvaa konvektiota. *Lorenz*-yhtälöt muodostavat seuraavanlaisen mallin: ilmakehässä lähellä maan pintaa oleva kaasu lämpenee, ja alkaa nousta ylöspäin. Noustessaan kaasu jäähtyy, ja ylhäällä se alkaa (painavampana/kylmempänä kuin ympäristönsä kaasu) vajota alaspäin. Näin syntyy konvektio, joka on tietyissä olosuhteissa hyvä esimerkki kaaoksesta ja kaoottisesta käyttäytymisestä. Tällä mallilla Lorenz osoitti, että hyvinkin yksinkertaisella yhtälöryhmällä voi olla ratkaisuja, joiden tarkkaa käyttäytymistä on mahdotonta ennustaa yhtälöitä ratkaisematta (tämä on ensimmäisiä tutkimuksia kaaoksesta).



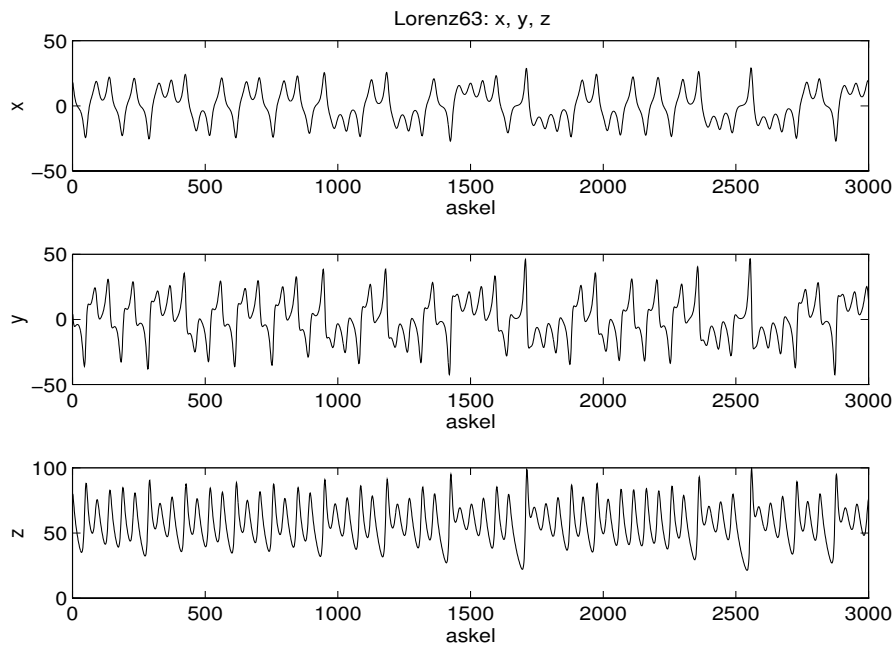
Kuva 4. *Lorenz* attractori (yhtälöryhmä 15 integroituna), parametrit ovat $p=10$, $r=63$ ja $b=8/3$.

Lorenz-yhtälöiden johtaminen alkaa *Navier-Stokesin* hydrodynaamiikan yhtälöistä; johto on melko pitkä ja se ei ole tässä yhteydessä oleellinen. Näinollen esitetään pelkkä ratkaisu:

$$\begin{aligned}
\dot{x} &= p(y - x) \\
\dot{y} &= rx - xz - y \\
\dot{z} &= xy - bz
\end{aligned}
\tag{15}$$

Tässä x kuvaa nesteen liikettä, y sekä z lämpötilaeroja. Pisteellä merkityt suureet ovat aikaderivaattoja. Mallin parametrit ovat p , r ja b , joiden tarkempi merkitys ei ole oleellinen. Valitaan yleisesti käytetyt parametrit: $p=10$, $r=63$ ja $b=8/3$, joita käytetään tästä eteenpäin kaikissa *Lorenz*-mallista tehdyissä ajoissa. Integroitaessa yhtälöitä 15 ajan mukana eteenpäin, saadaan x , y ja z ajan funktiona. Kun ne tulostetaan kolmiulotteiseen avaruuteen, nähdään kuva systeemin tilan ajan funktiona muodostamasta radasta. Tämä monille tutun näköinen kolmiulotteinen kuvaaja on kuvassa 4. Kuvassa 5. on nähtävissä muuttujien x , y ja z kehitys ajan funktiona.

Lorenz-malli on hyvin yleisesti käytetty kaottinen kuvaus, jota on tutkittu paljon. Yhtälöiden dynamiikka on hyvin monitahoista, ja se sisältää monenlaista käyttäytymistä riippuen malliin valituista parametreistä. Tämän takia se on hyvä vertailukohta tutkittaessa erilaisia analysointimenetelmiä.



Kuva 5. *Lorenz*-yhtälöiden (15) tuottamia aikasarjoja, muuttujien x , y ja z kehitys ajan funktiona ($p=10$, $r=63$ ja $b=8/3$).

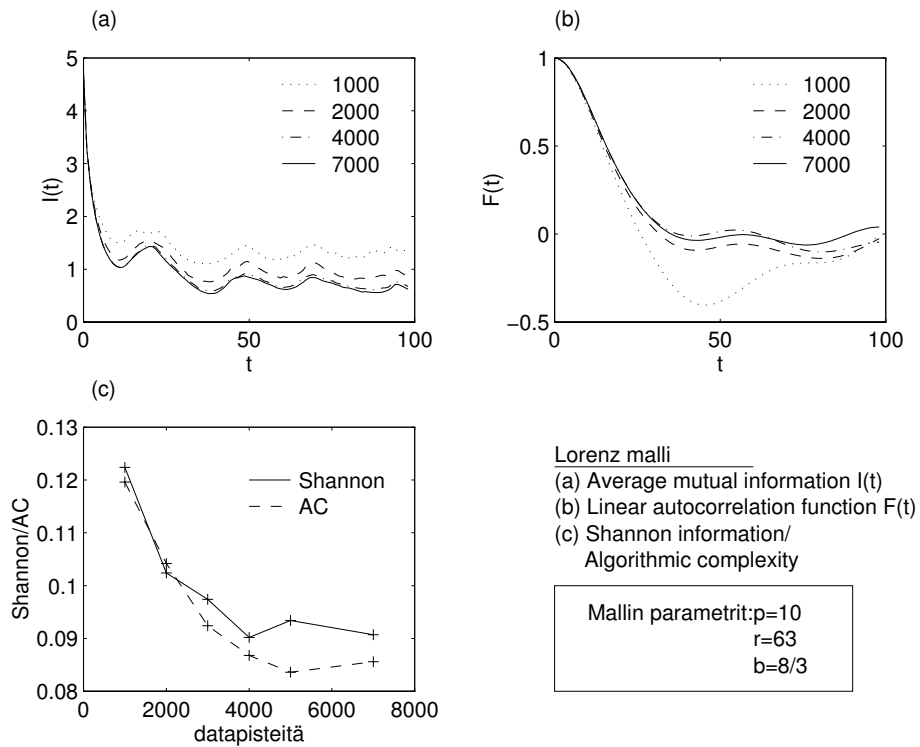
Koska yhtälöistä saatu aikasarja on jatkuvan analyttisen differentiaaliyhtälöryhmän tuottamaa, täytyy kappaleen 5 menetelmiä sovellettaessa hieman miettiä, kuinka aika-

sarjaa tulkitaan. Jatkuvuudesta johtuen nähdään, että peräkkäiset pisteet aikasarjassa korreloivat melko vahvasti (kuva 6, (a) ja (b)). Jos aikasarjasta lähdetään laskemaan sellaisenaan (oletus: $\delta = 1$) tila-avaruuden dimensiota d (FNN menetelmä, kappale 5.5), saadaan väärä tulos, koska tila-avaruuteen sijoitetut peräkkäiset pisteet korreloivat liian voimakkaasti keskenään. Yhtälöissä 9 - 12 käytetyn viiveparametrin δ (*time delay*) arvoa täytyy määrittää hieman tarkemmin.

Viiveparametrin δ arvon ei saa olla niin pieni, että peräkkäiset tila-avaruuden pisteet korreloivat liian voimakkaasti keskenään. Toisaalta, jos δ :n arvo on liian suuri, ei peräkkäisillä tila-avaruuden pisteillä ole ollenkaan tilastollista riippuvuutta. Eli arvo täytyy valita siten, että erotetaan *systemin dynamiikasta* lähtöisin olevat piirteet aikasarjan jatkuvuudesta.

Yksi tapa valita δ :lle sopiva arvo on tarkastella käsiteltävän aikasarjan *autokorrelaatiota*. Muissa tutkimuksissa on käytetty sääntönä parametrin arvolle sitä aikaa, jossa *lineaarinen autokorrelaatiofunktio* ensimmäisen kerran menee nollan alapuolelle (tässä tapauksessa olisi $\delta \approx 20-40$, kuva 6). Toisaalta *lineaarinen autokorrelaatiofunktio* on nimensä mukaisesti lineaarinen (kappaleet 5.1 ja 5.2) ja ei siten ole välttämättä paras mahdollinen analysoitaessa epälineaarisia aikasarjoja. Toinen, myös tässä yhteydessä käytetty määritelmä, on valita δ *average mutual information* $I(t)$ avulla siten, että se saa arvon jossa on $I(t)$:n ensimmäinen minimi. Kuvan 5 mukaan tässä tapauksessa $\delta \approx 10$ (Abarbanel et al. 1993).

Tila-avaruuden dimensiota d laskettaessa FNN-menetelmällä kaikilla datapisteiden määrillä (1000-7000 pistettä) saatiin d :lle arvoksi 3. Tämä on oikea arvo, kuten *Lorenz*-yhtälöiden määritelmästä (15) voidaan nähdä (kolme muuttujaa x , y ja z). Kannattaa muistaa tulevia kappaleita silmällä pitäen, että käytännössä, koska $\delta \approx 10$, esimerkiksi 1000 pistettä vastaa likimain 100:aa datapistettä (jolle asetetaan $\delta=1$)! Edelleen tarkasteltaessa kuvaa 6 nähdään, että kaikki käyrät (kuvat (a) ja (b)) konvergoivat likimain 4000:n datapisteen jälkeen. Eli jos datapisteiden lukumäärää nostetaan 4000:tta suuremmaksi, se ei enää muuta sanottavasti menetelmien antamia tuloksia. Tämä näkyy myös kuvan 6 osassa (c), jossa *Shannon-informaatio* ja *algorithmic complexity* konvergoivat niin ikään hieman 4000 datapisteen jälkeen.



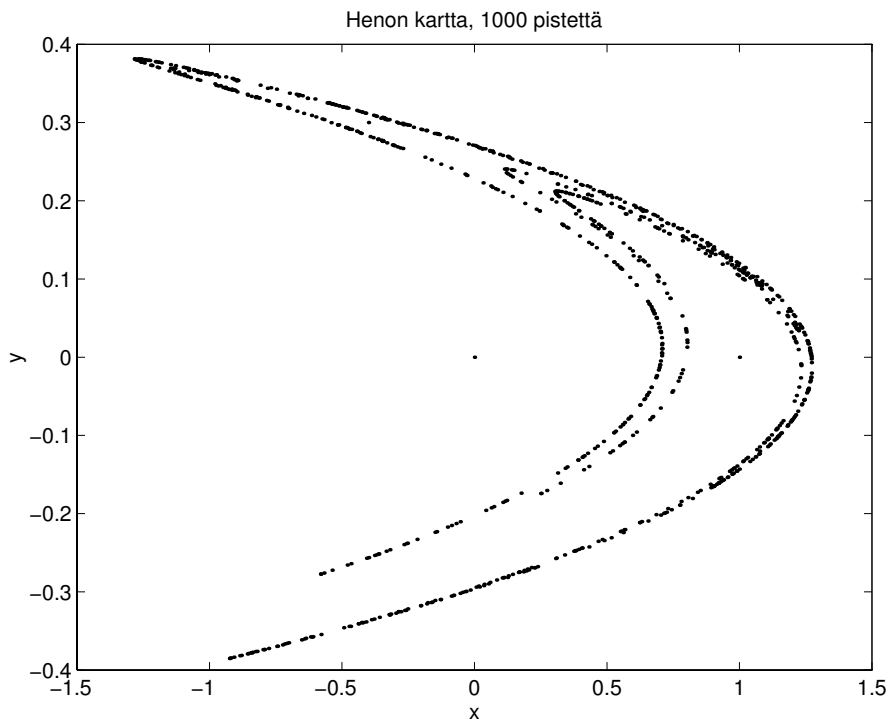
Kuva 6. *Lorenz*-yhtälöistä laskettuna (a) *average mutual information* 1000, 2000, 4000 ja 7000 datapisteellä, (b) *lineaarinen autokorrelaatiofunktio* 1000, 2000, 4000 ja 7000 datapisteellä sekä (c) *Shannon*-informaatio ja *algorithmic complexity* datapisteiden lukumäärän funktiona.

Kootaan vielä analysointimenetelmien antamat tulokset *Lorenz*-mallille:

Shannon-informaatio:	0.091
Algorithmic complexity:	0.085
Embedding dimensio d :	3

6.2 Hénon-kartta

Toinen paljon tutkittu kaottinen kuvaus on *Hénon*-kartta, jonka ensimmäisenä esitteli Hénon vuonna 1976. Se on kahden muuttujan rekursiivinen yhtälöpari (yhtälö 16). Tulostettaessa koordinaattien x ja y laskettuja arvoja saadaan kuvan 7 mukainen kuvaaja, josta nähdään, että näennäisesti satunnaisen aikasarjan (kuva 8) taustalla on selkeä järjestys ja dynamiikka.

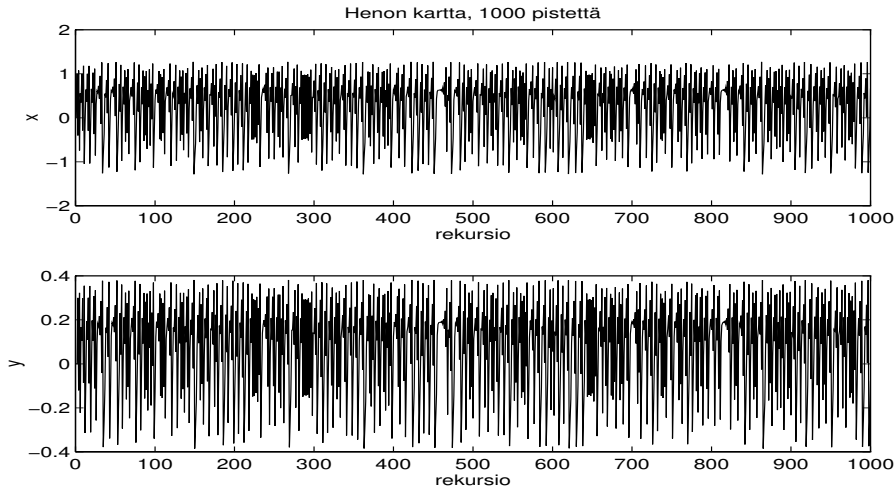


Kuva 7. *Hénon*-kartta. Parametrit ovat $a=1.4$ ja $b=0.3$.

Hénon-kartta määritellään seuraavan yhtälöparin avulla:

$$\begin{aligned}x_{n+1} &= 1 + y_n - ax_n^2 \\ y_{n+1} &= bx_n\end{aligned}\tag{16}$$

Tässä a ja b ovat kuvauksen parametrit. Yleisesti käytetyt arvot ovat $a=1.4$ ja $b=0.3$, joita käytetään kaikissa tässä raportissa esiintyvissä *Hénon*-karttaa koskevissa analyyseissä. Kuvassa 9 on esitetty yhteenveto kuvauksesta tehdyn analyysin tuloksista. Käytetyt menetelmät on käsitelty tarkemmin kappaleessa 5.

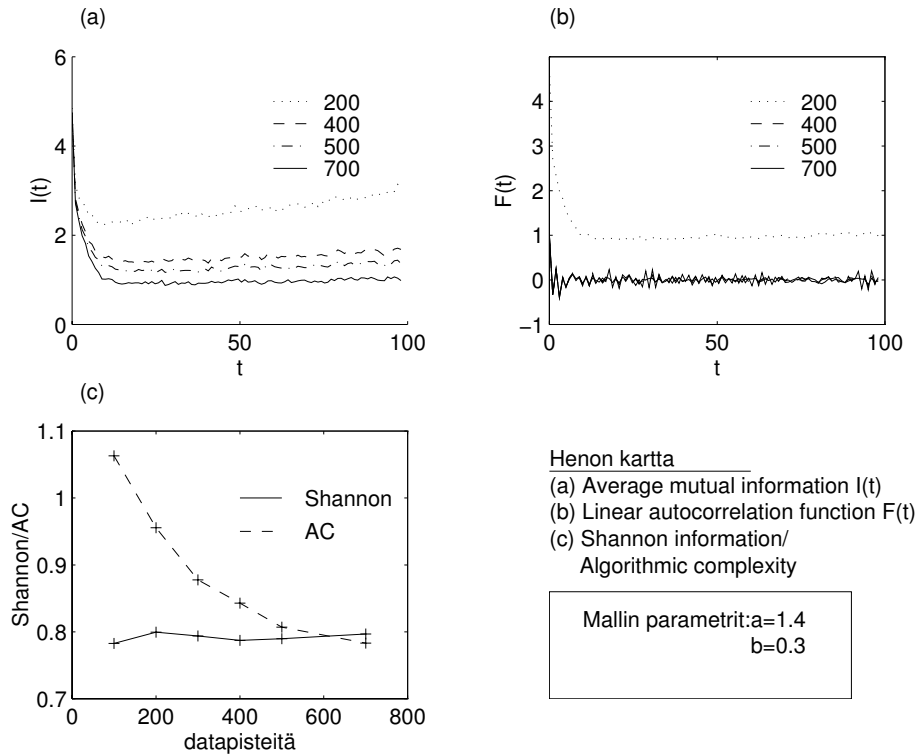


Kuva 8. Hénon-kartan avulla muodostettuja aikasarjoja. Kuvauksen parametrit ovat $a=1.4$ ja $b=0.3$.

Tutkittaessa kuvassa 9b esitettyä autokorrelaatiota nähdään, että lineaarinen autokorrelaatiofunktio ei osoita minkäänlaista aikasarjan sisäistä korrelaatiota. Ero on selkeä verrattaessa kuvan 6b Lorenz-yhtälöiden autokorrelaatiofunktioon: Hénon-kartalle $F(t)$ menee heti nolnaan ja jää sinne. Toisaalta *average mutual information* $I(t)$ osoittaa jonkinasteista sisäistä korrelaatiota tai muistia. Datapistemäärästä riippumatta $I(t)$ kääntyy tasaiseksi vasta, kun $t \approx 10$.

Kuvaajia 9a ja 9b tarkasteltaessa nähdään, että käyrät konvergoivat hieman 400 datapisteen jälkeen. Tämä on hieman tulkinnanvarainen tulos, joka riippuu siitä, kuinka tarkkoja halutaan olla. Myös kuvaajan 9c mukaan *Shannon-informaatio* ja *algorithmic complexity* ovat turvallisen lähellä toisiaan, kun datapisteitä on vähintään 400 pistettä.

Laskettaessa tila-avaruuden dimensiota d saatiin kaikilla datapisteiden määrillä $d=2$, mikä on oikea tulos (vrt. yhtälöpari 16). Tässä yhteydessä viiveparameterin δ valinnassa *average mutual information* käyrän avulla on myös tulkinnanvaraisuuksia; kuten kuvasta 9a nähdään, varsinaista minimiä ei ole. Kuvaaja lähestyy tiettyä arvoa tekemättä minimiä tms. Joka tapauksessa kokeilun tuloksena huomattiin, että viiveparameterin arvoksi kannattaa valita $\delta=1$.



Kuva 9. Hénon-kartasta laskettuna (a) *average mutual information* 200, 400, 500 ja 700 datapisteellä, (b) *lineaarinen autokorrelaatiofunktio* 200, 400, 500 ja 700 datapisteellä sekä (c) *Shannon-informaatio* ja *algorithmic complexity* datapisteiden lukumäärän funktiona.

Verrattaessa *Lorenz*-yhtälöiden ja *Hénon*-kartan tuottamia tuloksia keskenään nähdään, että ensimmäisen kompleksisuuden aste on toista matalampi; kuitenkin tila-avaruuden dimensio on *Lorenz*-yhtälöiden tapauksessa suurempi kuin *Hénon*-kartalla. Tämä johtuu siitä, että *Lorenz*-yhtälöt ovat analyyttinen differentiaaliyhtälöryhmä, ja *Hénon*-kartta diskreetti rekursiivinen kuvaus. Näinollen *Lorenz*-mallin yhteydessä *Shannon-informaatio* ja *algorithmic complexity* sekä sisäisen korrelaation mittarit saataisiin tarkemmin vastaamaan *Hénon*-kartasta saatuja tuloksia, jos niitä laskettaessa käytettäisiin samaa näyteenottotaajuutta kuin FNN-menetelmää sovellettaessa ($\delta=10$). Molemmissa tapauksissa menetelmät kuitenkin osoittavat aikasarjan sisäisten korrelaatioiden olemassaoloa, tulokset eivät vain ole sellaisenaan verrattavissa toisiinsa.

Mielenkiintoinen havainto on se, että jos *Lorenz*-yhtälöistä laskettu *Shannon-informaatio* ja *algorithmic complexity* kerrotaan käytetyllä viiveen δ arvolla, tulokset täsmäävät: *Lorenz*-yhtälöiden kompleksisuuden aste on *Hénon*-karttaa suurempi.

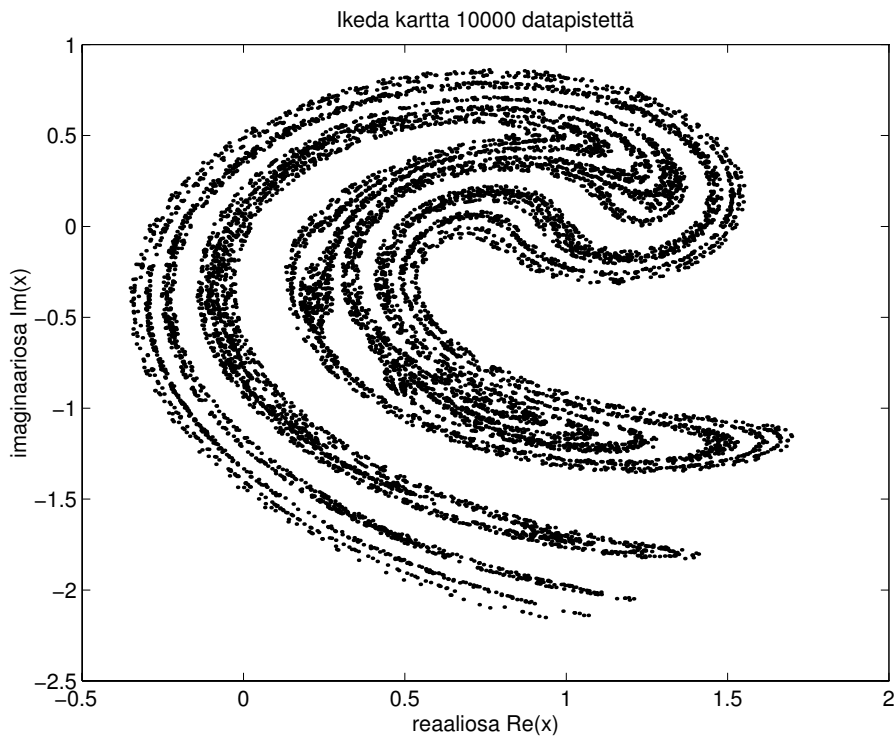
Kootaan vielä analysointimenetelmien antamat tulokset Hénon-kartalle:

Shannon-informaatio:	0.791
Algorithmic complexity:	0.811
Embedding dimensio d :	2

6.3 Ikeda-kartta

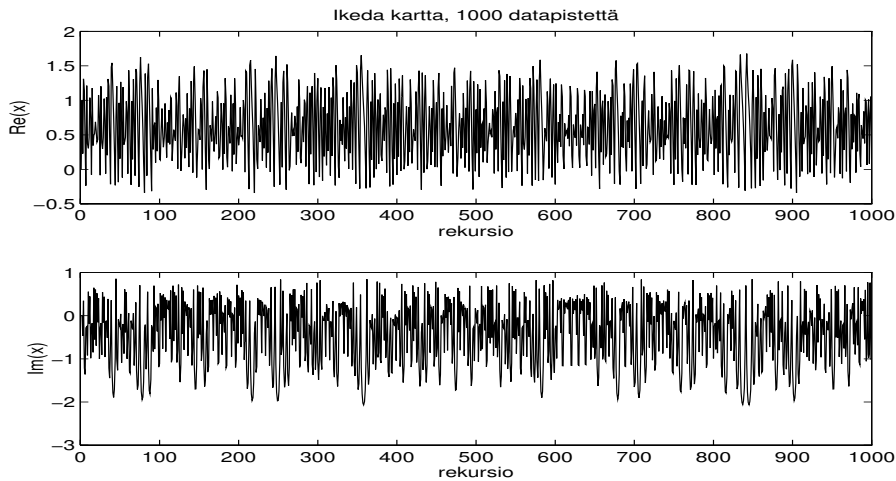
Ikeda esitteli vuonna 1979 kaaottisen kuvauksen, joka mallintaa lasersäteiden käyttäytymistä. Kuvauksen on kompleksitasolta $z(t)=x(t)+iy(t)$ takaisin kompleksitasolle:

$$z(t+1) = p + Bz(t)e^{i\kappa - \frac{i\alpha}{1+|z(t)|^2}}. \quad (17)$$



Kuva 10. Ikeda-kartta. Parametrit ovat $p=1.0$, $B=0.9$, $\kappa=0.4$ ja $\alpha=6.0$.

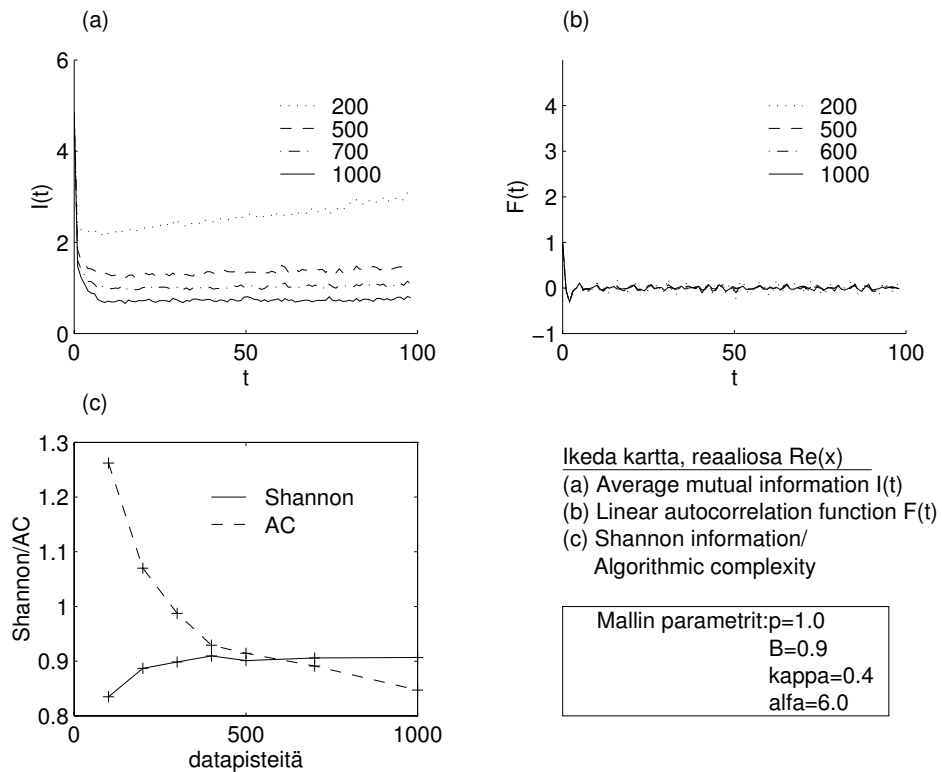
Tässä kuvauksen parametrit ovat p , B , κ ja α . Kirjallisuudessa yleisesti käytetyt arvot parametreille ovat $p=1.0$, $B=0.9$, $\kappa=0.4$ ja $\alpha=6.0$. Näitä arvoja käytetään myös tämän tutkimuksen yhteydessä käytettäessä *Ikeda*-kvausta. Kuvassa 10 on tulostettuna kuvauksen reaali- ja imaginääriosien muodostama kaksiulotteinen kuvaaja, josta näkee selvästi, että tuotetut aikasarjat (kuva 11) sisältävät rakennetta.



Kuva 11. *Ikeda*-kartasta tulostettuna reaali- ja imaginääriosia erikseen. Kuvauksen parametrit ovat $p=1.0$, $B=0.9$, $\kappa=0.4$ ja $\alpha=6.0$.

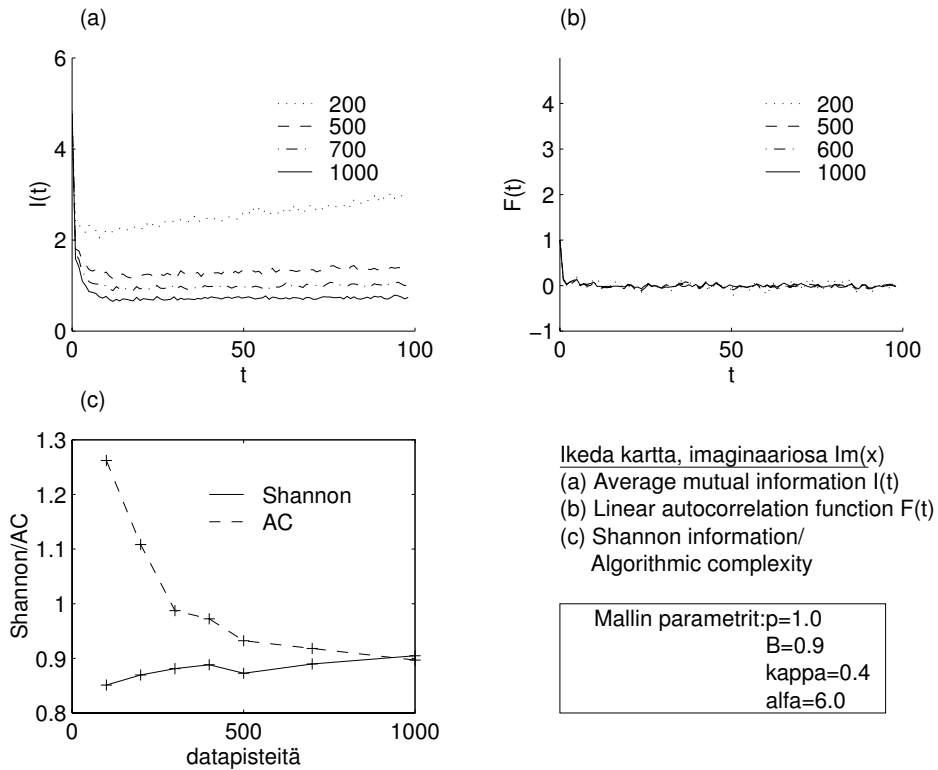
Kuvissa 12 ja 13 on *Ikeda*-kuvaukselle tehdyn kappaleen 7 mukaisen analyysin tulokset. Reaali- ja imaginääriosat on käsitelty erikseen. FNN-analyysiä tehtäessä käytettiin viiveparametrin arvoa $\delta=1$. Tutkittaessa reaali- ja imaginääriosien *average mutual information* kuvaajia (kuvat 12a ja 13a) nähdään, että molemmissa tapauksissa $I(t)$ kääntyy tasaiseksi, kun $t \approx 10$. Kuvaajissa ei ole nähtävissä varsinaisia minimejä, ja lyhyen kokeilun tuloksena huomattiin, että ainoa käyttökelpoinen viiveparametrin δ arvo on 1 (vrt. *Hénon*-kartan viiveparametri kappale 6.2).

Tarkasteltaessa analyysissä laskettujen arvojen konvergoitumista nähdään, että imaginääri- ja reaaliosilla tulokset alkavat olla suhteellisen lähellä toisiaan, kun datapisteitä on enemmän kuin 400 kappaletta (kuten kaikissa muissakin tapauksissa, kappaleet 6.1 ja 6.2). *Lineaarinen autokorrelaatiofunktio* ei näytä minkäänlaista aikasarjan sisäistä korrelaatiota kummassakaan tapauksessa (kuvat 12 ja 13).



Kuva 12. Ikeda-kartan reaali-osa laskettuna (a) *average mutual information* 200, 500, 700 ja 1000 datapisteellä, (b) *lineaarinen autokorrelaatiofunktio* 200, 500, 700 ja 1000 datapisteellä sekä (c) *Shannon-informaatio ja algorithmic complexity* datapisteiden lukumäärän funktiona (vertaa imaginaari-osa, kuva 13).

Taulukkoon 2 on koottu FNN-algoritmin tuottamat tulokset. Reaali-osalle $d=4$ yhtä poikkeusta lukuunottamatta. Imaginaari-osalle $d=3$. Kirjallisuudessa annetaan Ikeda-kartan *embedding dimensioksi* 4.



Kuva 13. *Ikeda*-kartan imaginääriosasta laskettuna (a) *average mutual information* 200, 500, 700 ja 1000 datapisteellä, (b) *lineaarinen autokorrelaatiofunktio* 200, 500, 700 ja 1000 datapisteellä sekä (c) *Shannon*-informaatio ja *algorithmic complexity* datapisteiden lukumäärän funktiona (vertaa reaaliosta, kuva 12).

Taulukko 2. FNN-menetelmällä lasketut tila-avaruuden dimensiot d *Ikeda*-kartan reaali- ja imaginääriosasta. Symboli ' ∞ ' merkitsee sitä, että FNN ei kyennyt määrittämään dimensiota alle $d=100$.

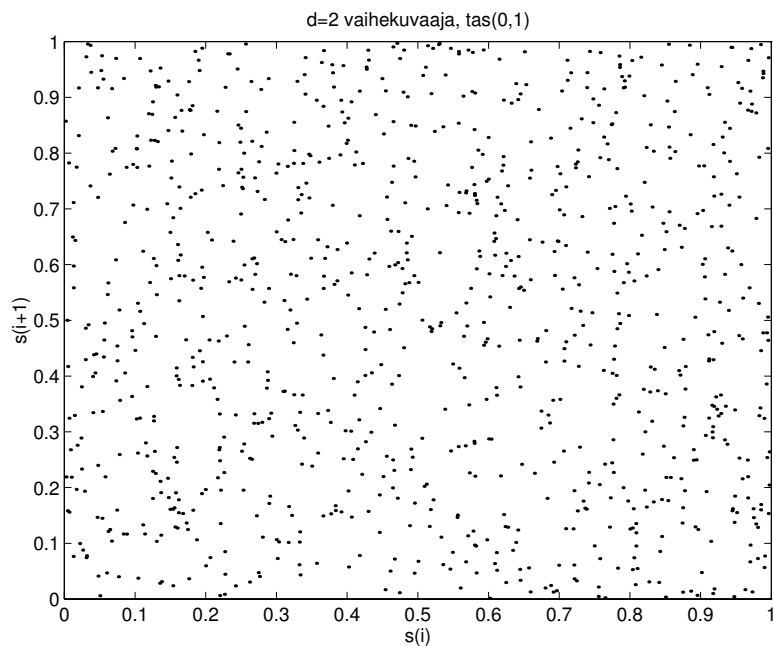
Datapisteitä	d reaaliosta $Re(x)$	d imaginääri- osasta $Im(x)$
100	∞	∞
200	∞	3
300	∞	3
400	4	3
500	3	3
700	4	3
1000	4	3

Kootaan vielä analysointimenetelmien antamat tulokset *Ikeda*-kartalle:

Reaaliosan Shannon-informaatio:	0.906
Reaaliosan algorithmic complexity:	0.895
Reaaliosan embedding dimensio d :	4
Imaginääriosan Shannon-informaatio:	0.889
Imaginääriosan algorithmic complexity:	0.930
Imaginääriosan embedding dimensio d :	3

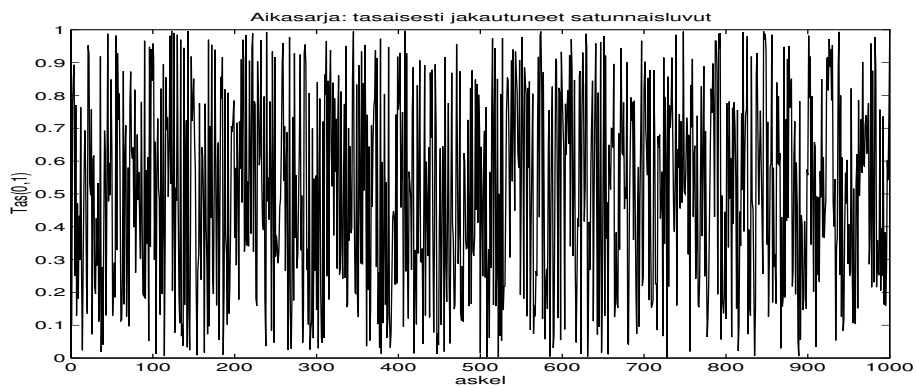
6.4 Kohina

Viimeinen ja ehkä tärkein testi tapahtuu pelkillä satunnaisluvuilla, joilla ei ole sisäistä korrelaatiota. Analyysissä käytetään kuvan 15 mukaista satunnaista, tasaisesti jakautunutta aikasarjaa s_i . Kuvasta 14, johon on tulostettuna kahteen ulottuvuuteen siirretty tilakuvaaja satunnaisluvuista, nähdään, että pisteet ovat yhä edelleen satunnaisia ja tasaisesti jakautuneita. Kuvassa 16 on kooste kappaleessa 5 esitettyjen menetelmien mukaisen analyysin tuloksista.

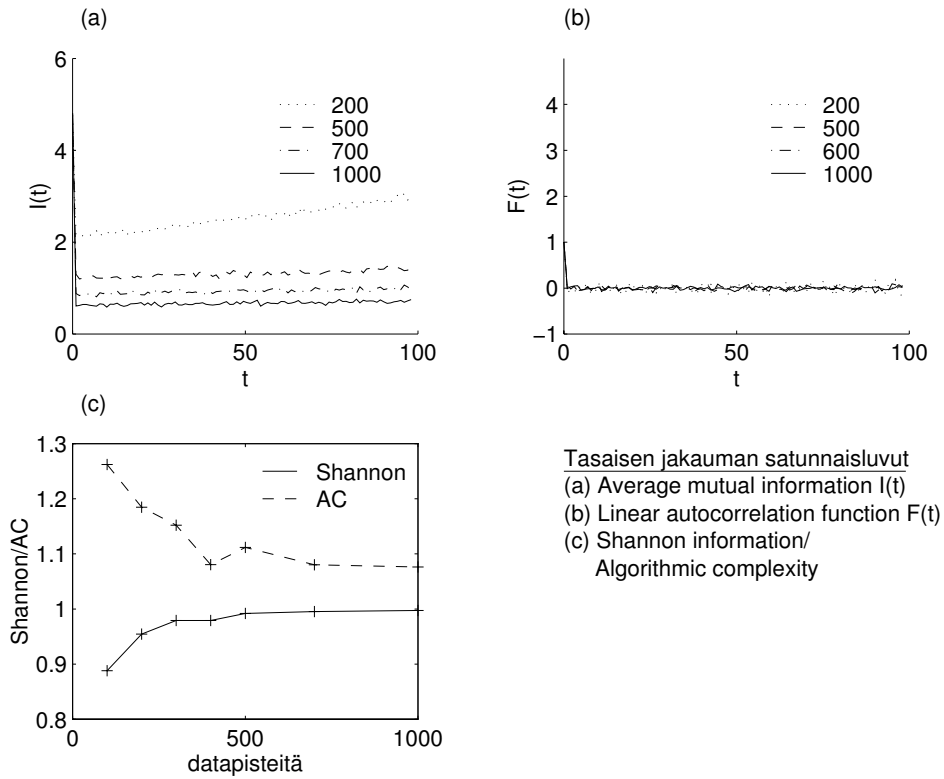


Kuva 14. Satunnaisluvut tulostettuna s.e. x-koordinaatti on piste s_i ja y-koordinaatti s_{i+1} .

Periaatteessa kohina sisältää “maksimimäärän” informaatiota. Koska ei ole olemassa sääntöä, jolla täysin satunnaisen aikasarjan voisi luoda uudelleen, täytyy täsmällisen kopion tekemiseen tietää jokaisen pisteen tarkka sijainti. Näinollen, kuten myös kuva 16c osoittaa, lähestyy *Shannon-informaatio* ja *algorithmic complexity* “maksimi” arvoaan 1.0. Lisäksi aikasarjan sisäistä korrelaatiota mittaavien *average mutual information* ja *lineaarisen autokorrelaatio* kuvaajan täytyy mennä heti lähelle nollaa.



Kuva 15. Esimerkki analyysissä käytetystä aikasarjasta, joka koostuu tasaisesti jakautuneista satunnaisluvuista. Vertaa kappaleiden 6.1, 6.2 ja 6.3 aikasarjojen kuvaajat (kuvat 5, 8 ja 11).



Kuva 16. Tasaisesti jakautuneista satunnaisluvuista muodostetusta aikasarjasta laskettuna (a) *average mutual information* 200, 500, 700 ja 1000 datapisteellä, (b) *lineaarinen autokorrelaatiofunktio* 200, 500, 700 ja 1000 datapisteellä sekä (c) *Shannon-informaatio* ja *algorithmic complexity* datapisteiden lukumäärän funktiona

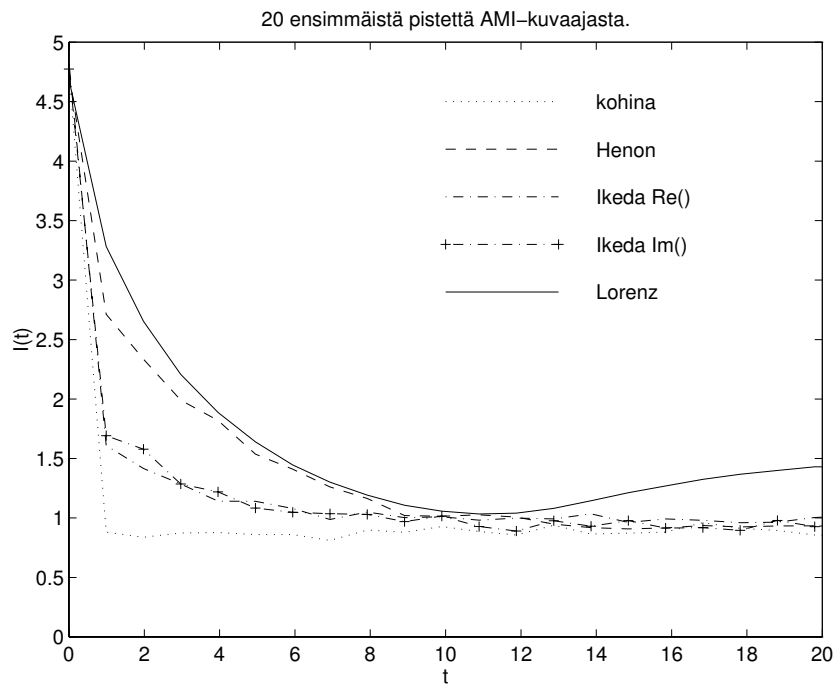
Laskettaessa FNN-menetelmällä satunnaisesti jakautuneella aikasarjalla tila-avaruuden dimension d arvoa, saatiin kaikilla datapisteiden määrillä odotettavissa ollut tulos: $d \rightarrow \infty$ eli/tai dataa ei ole tarpeeksi. Viiveparametrille käytettiin arvoa $\delta=1$. Myös datapisteiden lukumäärän kriteeri säilyy edellisten esimerkkien kaltaisena; 400 datapisteen jälkeen menetelmien antamat tulokset eivät muutu merkittävästi.

Kootaan vielä analysointimenetelmien antamat tulokset tasaisesti jakautuneelle satunnaiselle aikasarjalle:

Shannon-informaatio:	0.995
Algorithmic complexity:	1.089
Embedding dimension d :	∞

6.5 Esimerkkitapausten tulkinta

Edellisissä kappaleissa käsiteltyjen esimerkkitapausten avulla pyritään osoittamaan, että kappaleessa kuusi esitellyillä menetelmillä voidaan erottaa aikasarjasta onko se matalasti vai korkeasti kompleksinen. Tässä kappaleessa tehdään yhteenveto esimerkkitapausten saaduista tuloksista ja analysoidaan niitä hieman.



Kuva 17. Eri esimerkkitapausten *average mutual information* kuvaajien (AMI) ensimmäiset 20 pistettä.

Kun esimerkkitapaukset asetetaan järjestykseen niiden kompleksisuuden mukaan pienemmästä suurempaan, saadaan taulukon 3 mukainen lista. Varmien merkkien kompleksisuuden tasosta on tila-avaruuden dimensio d , jota on myös käytetty kriteerinä verrattaessa kompleksisuuksia keskenään. Eli, jos d saa FNN-menetelmällä kohtuullisen pienen arvon (esim. alle 10), on aikasarja melko varmasti mallinnettavissa ja ennustettavissa jollain aikavälillä. d :n arvo ' ∞ ' merkitsee sitä, että saatu dimensio on liian suuri ($d > 100$ FNN-algoritmin tuloksena). Tämä johtuu yleensä siitä, että dataa on liian vähän tai datan kompleksisuus on liian korkea.

Kirjallisuudessa on esitetty useita erilaisia arvioita vähimmälle tarvittavalle datapisteiden lukumäärälle aikasarjoja analysoidessa (FNN) embedding analyysillä (Abarbanel

et al. 1993). Kriteerit ovat yleensä sidoksissa aikasarjan tila-avaruuden dimensioon d ; tavallisesti käytetty likimääräinen kriteeri on

$$N = 10^d. \quad (18)$$

Tämän mukaan esimerkkitaapauksissa vaadittu datapisteiden määrä vaihtelee 100 pisteestä 10000 pisteeseen! Kohinan tapauksessa pisteitä ei voi olla koskaan riittävästi. Kuitenkin käytännön laskuissa nähdään, että esimerkkitaapauksissa 400 datapistettä on riittävästi kaikkien analyysien tekemiseen kohtuullisella tarkkuudella (kuvat 6, 9, 12, 13 ja 16). Kriteerinä käytännön laskuissa datapisteiden lukumäärälle käytetään sitä datapisteiden lukumäärää, jonka jälkeen menetelmien antamat tulokset eivät muutu merkittävästi sitä kasvatettaessa. Kuitenkin datapisteiden suurempi määrä merkitsee aina varmempia tuloksia.

Taulukko 3. Kooste analyysimenetelmien antamista tuloksista eri kuvauksien osalta AC on *algorithmic complexity*, d on FNN-menetelmällä lasketun tila-avaruuden dimension arvo, AMI on *average mutual information* ja ACF on *(linear) autocorrelation function*.

Kuvaus	Shannon	AC	d	AMI	ACF
Henon	0.791	0.811	2	≈ 10	≈ 1
Lorenz	0.091	0.085	3	≈ 10	≈ 40
Ikeda Im()	0.889	0.930	3	≈ 10	≈ 1
Ikeda Re()	0.906	0.895	4	≈ 10	≈ 1
Kohina	0.995	1.089	∞	≈ 1	≈ 1

Tarkasteltaessa aikasarjojen sisäisiä korrelaatioita nähdään, että *lineaarinen autokorrelaatiofunktio (ACF)* kykenee erottamaan sisäisen korrelaation ainoastaan *Lorenz-kuvauksen* tapauksessa. ACF:n epälineaarinen yleistys *average mutual information (AMI)* sen sijaan selkeästi erottelee (kuva 17) korkean ja matalan kompleksisuuden aikasarjat toisistaan. Kuvasta 17 nähdään, että *Henon-* ja *Lorenz-kuvaukset*, jotka ovat matalasti kompleksisia, sisältävät myös voimakkaimmat sisäiset korrelaatiot. Myös *Ikeda-kuvauksen* reaali- ja imaginaariosat eroavat sekä kohinasta että *Henon-* ja *Lorenz-kuvauksista*. Selkeästi näkyy myös se, että kohinalla ei ole mitään sisäistä rakennetta: AMI laskee heti nolleen, ja pysyy siellä.

Shannon-informaatio ja algorithmic complexity (AC) ovat toisiaan vastaavia suureita. Taulukosta 3 nähdään, että molempien edelläolevien suureiden lähestyessä arvoa yksi on kyse kohinasta (korkeasti kompleksisesta aikasarjasta). Mitä yksinkertaisempi signaali, sitä pienemmän positiivisen arvon *Shannon-informaatio* ja *algorithmic complexity* saavat. Taulukossa näkyy poikkeuksena *Lorenz-kuvauksen* arvot, jotka ovat melko tarkasti 10 kertaa pienempiä, kuin niiden pitäisi olla. Tämä johtuu siitä, että *Lorenz-yhtälöille* suureita laskettaessa jouduttiin käyttämään 10 kertaa enemmän datapisteitä kuin muille testitapauksille. Tämä sen takia, että testiaikasarjaan saataisiin tarpeeksi monta jaksoa. Jos saadut tulokset kerrotaan kymmenellä, arvot täsmäävät (katso kappale 6.2). Lisäksi testitapauksista on havaittavissa hyödyllinen piirre: usein,

kun *Shannon-informaatio* ja *algorithmic complexity* ovat kohtuullisen lähellä toisiaan, datapisteitä on riittävästi analyysin tekemiseen.

Näinollen voimme tehdä seuraavat yleistykset, joita käytetään komponenttikannan analysoinnissa.

Matalan kompleksisuuden aikasarjalle:

- Usein *Shannon-informaatio* ja *algorithmic complexity* saavat arvot, jotka ovat vähemmän kuin 0.9 (tai lähellä sitä) ja kohtuullisen lähellä toisiaan (keskinäinen erotus on vähemmän kuin 0.1).
- Average mutual information laskee loivasti vakiotasolle (vrt. *Lorenz-*, *Henon-* ja *Ikeda-*kuvaukset, kuva 17).
- Tila-avaruuden dimensio d voidaan laskea, ja saadaan tulos joka on alle 10. Tämä on varmin osoitus aikasarjan rakenteesta.

Korkean kompleksisuuden aikasarjalle:

- Usein *Shannon-informaatio* ja *algorithmic complexity* saavat arvot, jotka ovat selkeästi enemmän kuin 0.9 (lähellä arvoa 1.0) ja kohtuullisen lähellä toisiaan (keskinäinen erotus on vähemmän kuin 0.1).
- Average mutual information laskee HETI vakiotasolle (vrt. kohina, kuva 17).
- Tila-avaruuden dimensio d antaa tulokseksi ' ∞ '.

Dataa ei ole tarpeeksi:

- *Shannon-informaatio* ja *algorithmic complexity* saavat arvot, jotka eroavat selkeästi toisistaan (erotus luokkaa 0.2-0.4 tai enemmän).
- Tila-avaruuden dimensio d antaa tulokseksi ' ∞ '.

Kuten ylläolevista kriteereistä nähdään, tapaukset "ei tarpeeksi dataa" ja "korkeasti kompleksista dataa" menevät osittain päällekkäin.

7 Komponenttikannan analysointi

Komponenttikannasta tehtiin analyysi kappaleessa kuusi kuvatuille menetelmillä. Tässä kappaleessa esitetään analyysin tulokset ja kuvataan analysoinnissa käytetty algoritmi sekä datan esikäsittely.

7.1 Analyysi

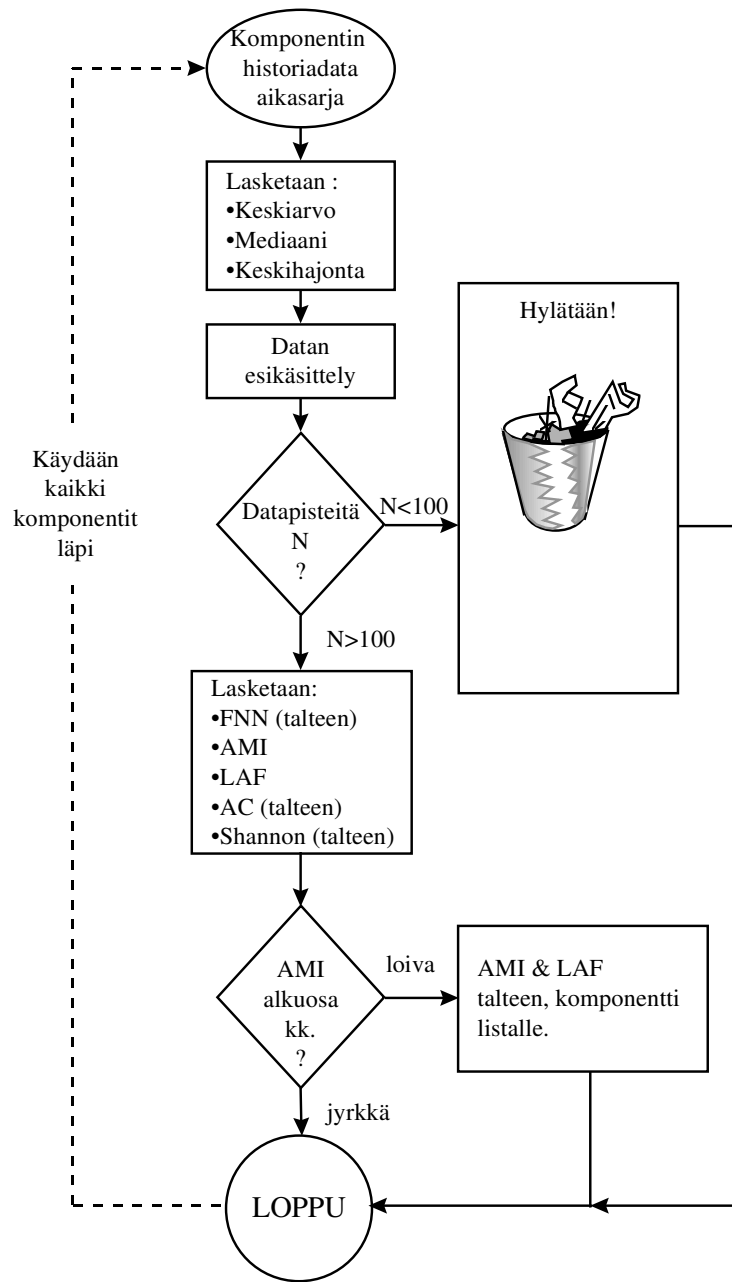
Kaikille kappaleessa 3 esitellyn haun tuloksena saaduille komponenteille tehtiin kuvassa 18 kuvattu analyysi.

Kuten kuvasta 18 nähdään, komponentin historiatiedoista muodostetun aikasarjan analyysi alkaa keskiarvon, mediaanin ja keskipoikkeaman laskemisella. Nämä lasketaan suoraan esikäsittlemättömästä datasta, ja ne kirjoitetaan kaikille komponenteille talteen. Tässä vaiheessa datapisteiden lukumäärä ei vaikuta komponenttien käsitteilyyn.

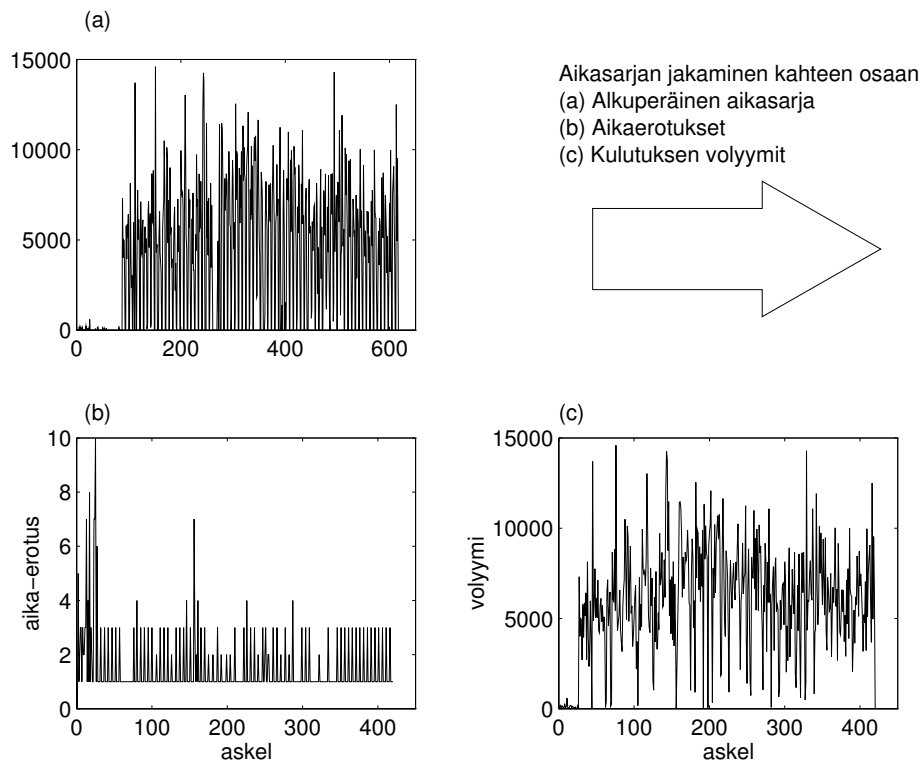
Keskiarvojen ja muiden vastaavien suureiden laskemisen jälkeen data esikäsitellään. Esikäsitteilyyn kuuluu kolme vaihetta:

1. Poistetaan kaikki datapisteet, joiden arvo on nolla.
2. Poistetaan kaikki reilusti muita suuremmat yksittäiset datapisteet (piikit).
3. Normalisoidaan data

Nollakulutuspäivät ovat pyhäpäiviä yms. yleisiä vapaapäiviä, jolloin tehtaalla ei ole ollut tuotantoa. Toinen mahdollinen syy nollakulutukselle on se, että sellaista tuotetta johon vaaditaan kyseistä komponenttia, ei ole ollut tuotannossa sinä päivänä. Lähes kaikkien komponenttien kulutushistoriatiedoissa viikon mittainen jakso näkyy esikäsittlemättömässä datassa hyvin voimakkaana ja peittää alleen muut datan piirteet. Poistamalla datasta kaikki nollakulutuspäivät voidaan ongelma poistaa tai sen vaikutusta vähentää. Tarkalleen ottaen aikasarja jaetaan kahteen osaan: *kulutuksen volyymi* ja *aika edelliseen kulutukseen* (kuva 19). Mikäli *kulutuksen volyymiä* oltaisiin kyetty ennakoimaan, olisi ennakoimista varten jouduttu analysoimaan myös aikaerotuksista syntyneitä aikasarjaa. Tässä tapauksessa se ei ole tarpeellista, joten keskitytään *kulutuksen volyymin* analysointiin.



Kuva 18. Komponenttikannalle tehdyn analyysin vuokaavio.



Kuva 19. Aikasarjan esikäsittelyssä data jaetaan kahteen osaan: komponenttien kulutuksen väliset ajat (peräkkäisten nollasta poikkeavien datapisteiden välinen aika) ja kulutuksen volyyymi (tällöin kulutettujen komponenttien lukumäärä).

Datassa olevien piikkien poistaminen on välttämätöntä sen takia, että ne hämäävät laskualgoritmeja ja johtavat väriin lopputuloksiin. Esimerkiksi lisättäessä kohinaan yksi reilusti yli 10 kertaa keskipoikkeaman korkuinen piikki, saadaan tila-avaruuden dimensioksi FNN-algoritmillä alle kymmenen (mikä on väärä vastaus). Aikasarja siivotaan siten, että käydään se läpi ensin kerran, ja poistetaan kaikki datapisteet joiden arvo on suurempi kuin aikasarjan keskiarvon ja kaksi kertaa keskipoikkeaman summa. Kun kaikki ko. ehdon täyttävät datapisteet on poistettu, käydään aikasarja läpi uudelleen vastaavalla tavalla uusilla (jo kerran “siistityillä”) keskiarvon ja keskipoikkeaman arvoilla. Tässä on kuitenkin tulkinnanvaraisuudesta johtuva ongelma; mitkä piikit on laskettava aikasarjaan ja mitkä ylimääräiseksi häiriöksi? Piikkien poiston vaikutusta testattiin hieman Hénon- ja Ikeda-kuvauksilla sekä komponenttikannalla (seuraava kappale).

Esikäsittelyn viimeisessä vaiheessa data normalisoidaan välille $[0,1]$.

Esikäsitellyn jälkeen tarkastellaan mitä datasta on jäänyt jäljelle: jos datapisteitä on yhä enemmän kuin 100 kappaletta, kyseinen komponentti hyväksytään jatkokäsittelyyn. Keinotekoisia testitapauksia käsiteltäessä nähtiin, että 400 pistettä on alaraja datapisteiden lukumäärälle, jotta tulokset olisivat edes jossain määrin luotettavia. Valittu 100 pistettä on reilusti alle tämän rajan, mutta se mahdollistaa suuremman komponenttijoukon käsittelyn. Tämä kannattaa pitää mielessä tuloksia tarkasteltaessa. Yli 400 datapisteen komponentteja on hyvin vähän (kappale 4). Hylätyn komponentin *Shannon-informaatiolle*, *algorithmic complexitylle* ja *tila-avaruuden dimensiolle* annetaan arvot -1, ja siirrytään käsittelemään seuraavaa komponenttia.

Seuraavana lasketaan komponentin esikäsitellystä historiadatasta *Shannon-informaatio* (Shannon), *algorithmic complexity* (AC), *tila-avaruuden dimensio* (d), *lineaarinen autokorrelaatiofunktio* (LAF) ja *average mutual information* (AMI). Näistä kirjoitetaan aina talteen Shannon, AC ja d.

Viimeisenä testataan *average mutual information* (AMI) käyrän käyttäytymistä. Kuten kappaleessa 5 osoitettiin, kaoottisissa kuvauksissa AMI laskee loivasti ja satunnaisissa kuvauksissa heti vakiotasolle. Niinpä viimeisessä testissä tarkastellaan AMI-kuvaajan alkuosan (datapisteet 2-20) käyttäytymistä. Alkuosan alueelle sovitetaan suoran yhtälö PNS-menetelmällä, ja tarkastellaan saadun suoran kulmakerrointa. Jos kulmakerroin on pienempi kuin -0.005, tulkitaan lasku loivaksi. Tällöin komponentista laitetaan talteen AMI-kuvaaja ja LAF-kuvaaja. Tällaiset komponentit on listattu raportin lopussa olevassa liitteessä (liite 1).

Viimeisen vaiheen jälkeen otetaan tarkasteltavaksi seuraava komponentti, ja näin jatketaan kunnes kaikki tiedostossa olevat komponentit on käyty lävitse.

7.2 Analysoinnin tulokset

Tarkastellaan ensin datan esikäsitelyssä tehtävän “piikkien” poistamisen vaikutusta tuloksiin. Ulkoisten häiriöiden vaikutusta analysointimenetelmien toimintaan testattiin *Hénon*- ja *Ikeda*-kuvauksilla sekä satunnaisella aikasarjalla lisäämällä niihin tietty määrä kohinaa tai “piikkejä”.

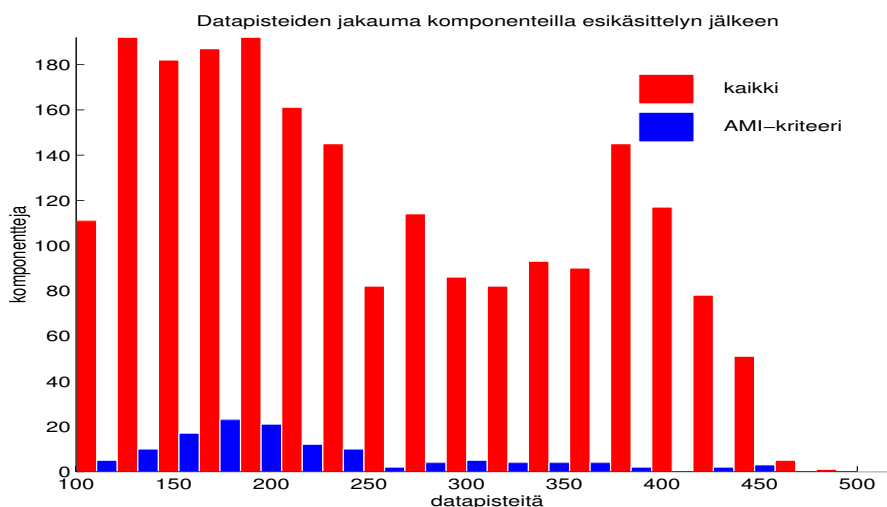
Testitapauksilla laskettaessa häiriöiden vaikutusta, käytettiin 400 datapisteen aikasarjoja. Tämä on lähellä sitä rajaa, jolla menetelmät alkavat tuottaa kelvollisia tuloksia. Kun aikasarjaan lisättiin 1-5 kappaletta piikkejä, korkeuksiltaan keskiarvoon lisätynä 2-5 kertaa aikasarjan keskihajonta, nähtiin seuraavat yleiset piirteet:

- Lisättyjen piikkien lukumäärä ei merkitse paljoa Shannon-informaatiota ja algorithmic complexitya laskettaessa, yksi piikki riittää muuttamaan tuloksia yhtä paljon kuin esimerkiksi viisi piikkiä.
- Lisättyjen piikkien korkeudet ovat merkittäviä: alussa piikkien lisääminen nosti Shannon/AC arvoja, mutta piikkien korkeutta lisättäessä edelleen arvot laskivat voimakkaasti. Tämä johtuu siitä, että ko. menetelmiä käytettäessä data jaetaan diskreetteihin symboleihin siten, että eri symbolit ovat tasaisesti lokeroituna aikasarjan maksimi- ja minimiarvon välille. Lisättäessä tarpeeksi korkea piikki dataan ainoastaan lisätty piikki pääsee ylempään diskreettiin arvoon, muut jäävät alem-

paan. Näin aikasarjan rakenne häviää diskreetissä kuvauksessa. Raja tälle tapahtumalle oli noin 3-4 keskipoikkeamaa.

- Tila-avaruuden dimension määrittämiseen käytetty FNN-algoritmi on hyvin herkkä yksittäisille piikeille. Jo yksi piikki riittää sotkemaan tulokset. Häiriön voimakkuus riippuu vahvasti piikkien lukumäärästä ja korkeuksista. Tämä voi osittain johtua myös siitä, että käytetty datapisteiden lukumäärä on liian alhainen FNN-algoritmile. Yhden kaksi kertaa keskipoikkema korkean piikin lisääminen riittää antamaan vääriä tuloksia.
- FNN-algoritmi kestää melko hyvin aikasarjaan lisättyä kohinaa; tulokset alkavat muuttua vasta, kun kohina on 10-20 % keskihajonnasta. Shannon-informaatiota laskettaessa kohinan lisäystä tuskin havaitaan, mutta algorithmic complexity alkaa selkeästi nousta kohinaa lisättäessä. Tämä voi olla osoitus AC:n kyvystä luonnehtia epälineaarisia aikasarjoja Shannon-informaatiota paremmin.

Nämä tulokset osoittavat selvästi, että aikasarjassa voimakkaat piikit täytyy poistaa dataa esikäsiteltäessä. Kriteerinä poistettavien piikkien korkeudelle on analyysiä tehdessä käytetty korkeutta, joka on aikasarjan keskiarvo ja siihen lisättyä kaksi kertaa aikasarjan keskihajonta. Ongelmana on kuitenkin piikkien poistamisen tuoma tulkinanvaraisuus rajatapauksissa. Käytännössä dataa analysoidessa huomataan, että kohuullisen usein data sisältää paria *kertaluokkaa* keskiarvoa korkeampia piikkejä!



Kuva 20. Komponenttien historiatietojen datapisteiden määrän jakauma datan esikäsitellyn jälkeen. Kuvassa näkyy kaikki komponentit ja ne komponentit erikseen, jotka ovat täyttäneet AMI-kriteerin ja päässeet dokumentin lopussa olevalle listalle.

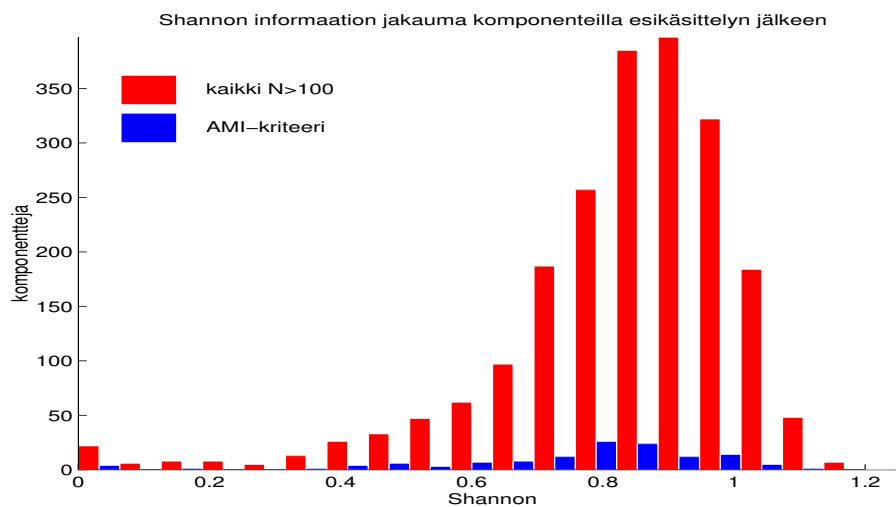
AMI-kriteerin analysoinnissa läpäisi 128 komponenttia, joka on noin 1.5 % kaikista (8332 kpl) ja 6.1 % esikäsitellyistä (2114 kpl) komponenteista. Keskimää-

räisen kriteerin täyttäneen komponentin historiatietoja oli 200 datapistettä. Taulukko 4 kuvailee keskimääräisen AMI-kriteerin täyttäneen komponentin ominaisuudet analysoinnissa.

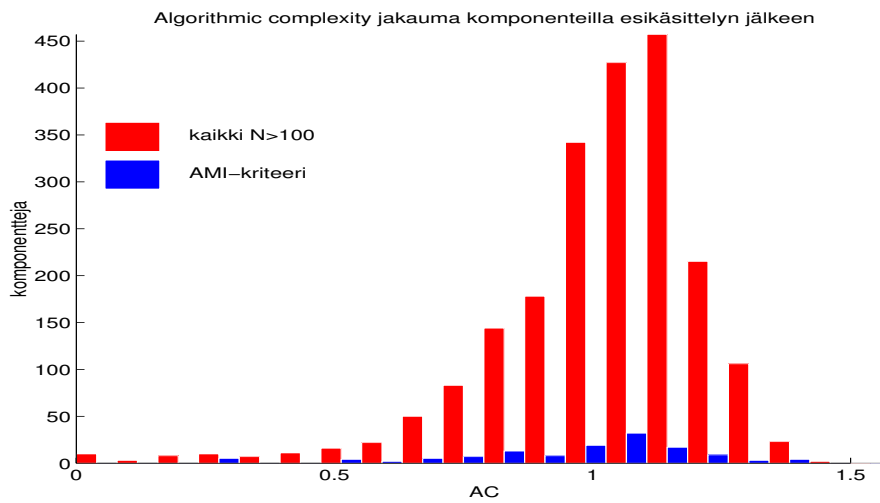
Taulukko 4. Keskimääräisen AMI-kriteerin täyttäneen komponentin saamat arvot analysoinnissa.

	Keskiarvo	Mediaani	Jakauman huippu
AC	0.9291	0.9924	≈1.1
Shannon	0.7258	0.7719	≈0.8
Datapisteitä	204	178	≈170

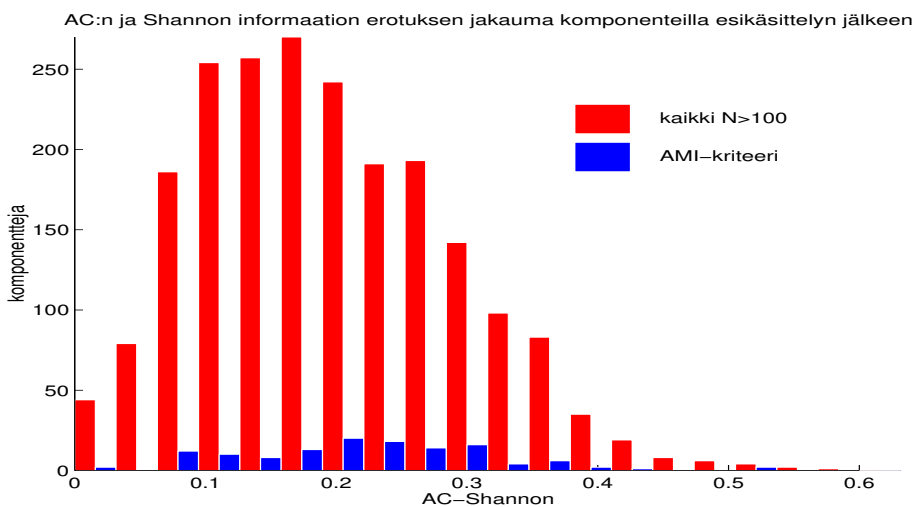
Vertailtaessa Shannon-informaation (kuva 21) algorithmic complexityn (kuva 22) ja datapisteiden lukumäärän (kuva 20) jakaumia AMI-kriteerin ja esikäsittelyn läpäisseiden komponenttien osalta nähdään, että jakaumat ovat likimain samanmuotoiset. Siten AMI-kriteeri ei ole sidoksissa merkittävässä määrin datapisteiden lukumäärään, Shannon-informaatioon tai algorithmic complexityyn. Ainoan poikkeuksen tässä tekee algorithmic complexityn ja Shannon-informaation välinen erotus AMI-kriteerin läpäisseillä komponenteilla, joka näyttää olevan yleistä jakaumaa enemmän painottunut suuremmille arvoille (kuva 23).



Kuva 21. Komponenttien historiatietojen Shannon-informaation jakauma datan esikäsittelyn jälkeen. Kuvassa näkyy kaikki komponentit ja ne komponentit erikseen, jotka ovat täyttäneet AMI-kriteerin ja päässeet dokumentin lopussa olevalle listalle.

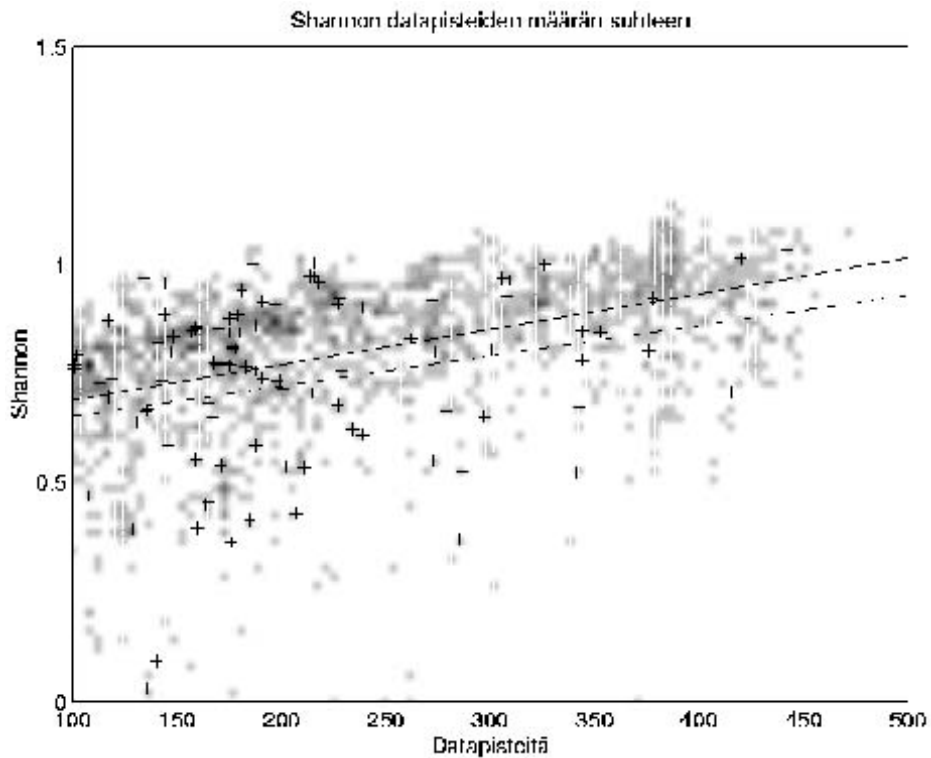


Kuva 22. Komponenttien historiatietojen algorithmic complexityn jakauma datan esikäsittelyn jälkeen. Kuvassa näkyy kaikki komponentit ja ne komponentit erikseen, jotka ovat täyttäneet AMI-kriteerin ja päässeet dokumentin lopussa olevalle listalle.



Kuva 23. Komponenttien historiatietojen algorithmic complexityn ja Shannon-informaation erotusten jakauma datan esikäsittelyn jälkeen. Kuvassa näkyy kaikki komponentit ja ne komponentit erikseen, jotka ovat täyttäneet AMI-kriteerin ja päässeet dokumentin lopussa olevalle listalle.

Analysoitaessa komponenttikantaa FNN-algoritmillä, jonka tuottamat positiiviset tulokset ovat varmin osoitus aikasarjan rakenteesta, nähdään, että kaikilla komponenteilla saadaan tuloksena $d=\infty$. Tämä viittaa siihen, että joko dataa on liian vähän ja/tai se ei sisällä rakennetta. Tapauksia ei voi erottaa toisistaan. Jos dataa esikäsiteltäessä piikkien poistamisessa käytettyä kriteeriä löysätään hieman, alkaa FNN-algoritmi tuottaa myös äärellisiä tuloksia. Tarkasteltaessa aikasarjoja, joille saadaan tulos $d \neq \infty$, nähdään, että niissä on tulokset sotkeva piikki. Kun piikki poistetaan, saadaan taas tulos $d=\infty$. Joka tapauksessa tulos on negatiivinen: aikasarja ei sisällä rakennetta joko sen takia, että sitä ei kertakaikkiaan ole olemassa tai sen takia, että dataa ei ole tarpeeksi.

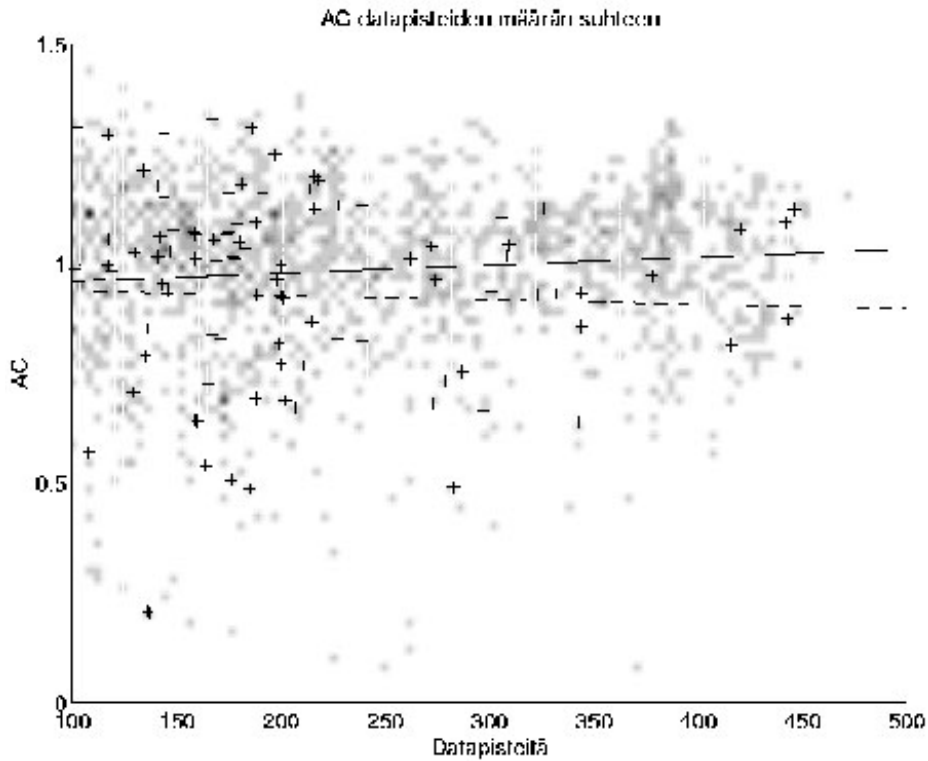


Kuva 24. Shannon-informaatio komponenttien datapisteiden määrän funktiona. Yksittäiset ristit ovat AMI-kriteerin läpäisseitä komponentteja, ja taustalla oleva harmaasävykartta kuvaa koko komponenttikantaa. Yhtenäinen viiva on PNS-sovitettu suora koko komponenttikantaan ja katkoviiva on vastaava sovitus AMI-kriteerin täyttäneisiin komponentteihin.

Tarkasteltaessa Shannon-informaation ja algorithmic complexityn käyttäytymistä datapisteiden määrän funktiona nähdään (kuva 24) tyypillisen Shannon-informaation arvon nousevan datapisteiden lukumäärän myötä. Kun datapisteiden lukumäärä lähestyy esimerkkitapauksissa määriteltyä rajaa (400 datapistettä), nähdään tyypillisen arvon lähestyvän 1.0:aa. Myös arvojen hajonta pienenee datapisteiden lukumäärän noustessa. Algorithmic complexity käyttäytyy vastaavasti (kuva 25) siten, että sen arvot ovat koko ajan datapisteiden lukumäärästä riippumatta lähellä 1.0:aa. Kuten Shannon-informaationkin tapauksessa arvojen hajonta pienenee selvästi datapisteiden lukumäärän kasvaessa. Tämä voidaan tulkita niin, että menetelmät antavat systemaattisesti virheellisiä arvoja (liian pieniä Shannon-informaatiolle ja liian suuria AC:lle) alle 400 datapisteen aikasarjoilla. Lisäksi lyhyt aikasarjadata on huomattavasti alttiimpi yksityiskohtien aiheuttamille virheille lopputuloksessa kuin pitkä. Tämä selittää tulosten jakauman kaventumisen datapisteiden lukumäärän kasvaessa. AC:n ja Shannon-informaation arvon lähestymisen 1.0:aa voidaan tulkita yhtäpitävänä havaintona FNN-algoritmin tuottavien tulosten kanssa; satunnaiselle aikasarjalle Shannon-informaatio ja AC saavat arvot lähellä 1.0:aa (kts. kappale 6.4).

Taulukko 5. AMI-kriteerin täyttäneiden komponenttien suhteellinen osuus kuillakin datapisteiden lukumäärillä. Kaikkiaan noin 6 % esikäsittelyn läpäisseistä komponenteista täyttävät AMI-kriteerin. Jos AMI-komponenttien jakauma olisi täsmälleen sama kuin kaikkien komponenttien jakauma, olisi suhteellinen osuus aina 6.0 %. Taulukosta nähdään, että AMI-komponenteista suurin suhteellinen osa on datapisteiden lukumäärällä 150-200.

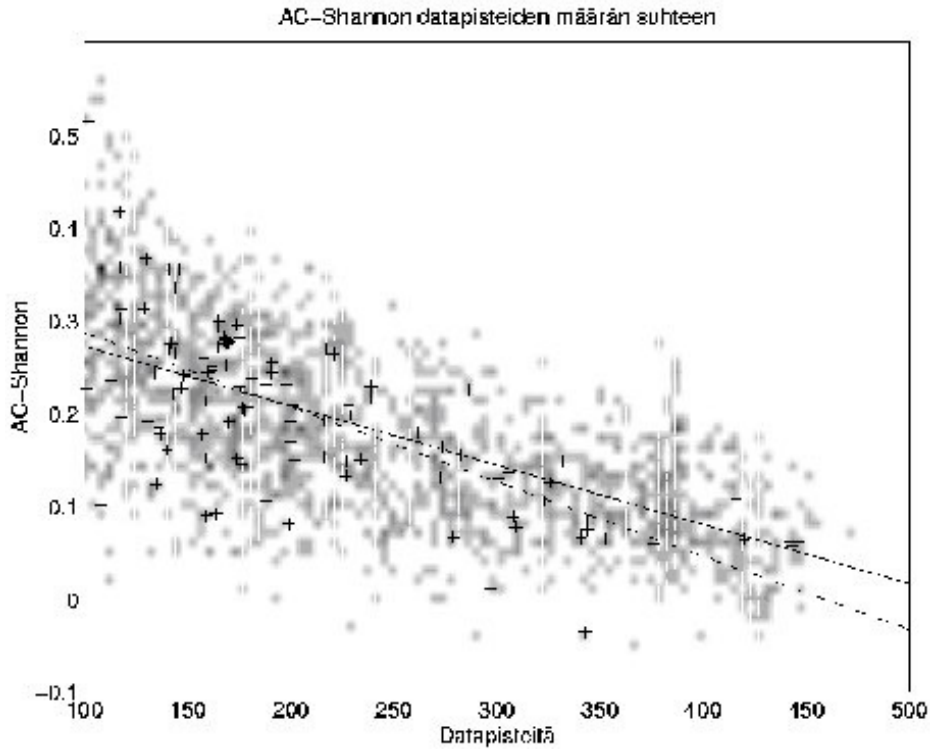
Datapisteitä	Komponentteja välillä	AMI-kriteerin täyttäviä komponentteja	AMI-komponenttien suhteellinen osuus kaikista komponenteista
100-150	483	32	6.6 %
150-200	449	47	10.5 %
200-250	315	21	6.7 %
250-300	224	9	4.0 %
300-350	208	11	5.2 %
350-400	301	3	0.9 %
400-450	151	5	3.3 %
450-	4	0	0.0 %



Kuva 25. Algorithmic complexity komponenttien datapisteiden määrän funktiona. Yksittäiset ristit ovat AMI-kriteerin läpäisseitä komponentteja, ja taustalla oleva harmaasävykartta kuvaa koko komponenttikantaa. Yhtenäinen viiva on PNS-sovitettu suora koko komponenttikantaan ja katkoviiva on vastaava sovitus AMI-kriteerin täyttäneisiin komponentteihin.

Jos tarkastellaan, kuinka AMI-kriteerin täyttävät komponentit ovat jakautuneet koko esikäsitellyssä komponenttikannassa, nähdään, että kriteerin täyttävät komponentit jakautuvat suhteellisesti lähes samalla tavalla kuin kaikki komponentit Shannon-informaation tai algorithmic complexityn ja datapisteiden lukumäärän suhteen (vertaa ristien ja harmaasävykartan jakaumaa sekä PNS-sovitettuja suoria kuvissa 24 ja 25). Myös AMI-komponenteilla näyttää olevan painopiste likimain 200 datapisteen kohdalla (taulukko 5). Tarkasti jakauman huippu on AMI-kriteerin täyttävillä komponenteilla (150-200 datapistettä) 50 datapistettä suurempi kuin kaikilla komponenteilla (100-150 datapistettä). Tämä on heikko osoitus siihen suuntaan, että analyysin tuloksena saatava komponenttien rakenne tai informaatioisältö riippuisi siitä kuinka paljon datapisteitä on käytettävissä. Eli, jos *analysointimenetelmät toimivat voimassaoloalueensa rajoilla JA komponenttien historiatatalla on rakenne*, se näkyisi siinä, että datapisteiden lukumäärän kasvaessa, myös suhteellinen AMI-kriteerin täyttävien komponenttien lukumäärä kasvaisi. Havaittu ero on todennäköisesti enemmän sattumaa kuin todellinen efekti.

Yksi vaihtoehto on, että AMI-kriteeri antaa pienillä datapisteiden arvoilla myös väärää tuloksia, ja näin vääristää AMI-komponenttien jakaumaa samassa suhteessa kaikkien komponenttien jakauman kanssa. Vertaa alhaisten datapisteiden määrillä lasketut testitapaukset kuvat 9 ja 12.



Kuva 26. Shannon-informaation ja algorithmic complexityn erotus komponenttien datapisteiden määrän funktiona. Yksittäiset ristit ovat AMI-kriteerin läpäisseitä komponentteja, ja taustalla oleva harmaasävykartta kuvaa koko komponenttikantaa. Yhtenäinen viiva on PNS-sovitettu suora koko komponenttikantaan ja katkoviiva on vastaava sovitus AMI-kriteerin täyttäneisiin komponentteihin.

Kuten kappaleen 6.5 loppuosassa mainittiin, näytti yksi kriteeri datapisteiden lukumäärälle ja tulosten luotettavuudelle olevan Shannon-informaation ja algorithmic complexityn välinen erotus. Tämä suure on tulostettuna komponenteille kuvassa 26, josta näkyy selkeästi kuinka erotus lähenee nollaa datapisteiden lukumäärän kasvaessa. Sama trendi on olemassa sekä kaikilla komponenteilla että AMI-kriteerin täyttävillä komponenteilla (vertaa kuvan 26 PNS-suoria toisiinsa). Kuvassa 23 on saman erotuksen histogrammi AMI-komponenteille ja kaikille komponenteille, josta nähdään kuinka suurin osa AMI-komponenteista on 200 datapisteen tienoilla ja näin AC-Shannon -erotukseltaan noin 0.2 tienoilla. Kuva 26 on osoitus siitä, että AC ja Shannon-arvot ovat luotettavia, vasta kun datapisteitä on noin 400 kappaletta. AC-Shannon -erotuksen arvoilla on hajontaa komponenttien keskuudessa myös 400 datapisteen jälkeen, mutta hajonta kapenee selvästi lähestyttäessä kyseistä rajaa.

8 Yhteenveto

Varsinaisen sovelluksen osalta tutkimuksen kaksi keskeisintä tulosta ovat:

- Dataa komponenttien kulutushistoriasta ei ollut saatavilla tarpeeksi pitkältä ajalta.
- Datan kompleksisuus on niin korkeaa, että vaadittujen ennusteajkojen saavuttaminen on mahdotonta.

Analysointimenetelmien tarkastelun kannalta tilanne on kuitenkin huomattavasti parempi:

- FNN-algoritmeilla, *algorithmic complexity*llä, Shannon-informaatiolla ja *average mutual information* (AMI) kuvaajalla voidaan erottaa satunnainen aikasarja matalasti kaoottisesta, jos dataa on tarpeeksi.
- FNN-algoritmin mukaan *komponenttien historiadata on satunnaista tai sitä ei ole yhdenkään komponentin kohdalla tarpeeksi luotettavan analyysin suorittamiseen*.
- AC- ja Shannon-arvot lähestyvät luotettavimmilla datapisteiden lukumäärillä arvoa 1.0 ja toisiaan (muutamia poikkeuksia on), mikä vastaa testitapauksissa satunnaista aikasarjaa. Chaitin osoitti vuonna 1987, että Markov prosesseille (satunnaisprosessi, jonka seuraavan arvon todennäköisyys riippuu viimeisimmistä tilasta) *algorithmic complexity* ja *Shannon-informaatio* käyttäytyvät samalla tavalla. Lisäksi suureet on normitettu niin, että satunnaisella aikasarjalla pitäisikin tulla tulokseksi likimain 1.0.
- AMI-komponentit ovat alhaisilla datapisteiden lukumäärillä, mikä viittaa lähinnä siihen, että *average mutual information* ei välttämättä anna luotettavia tuloksia lyhyillä aikasarjoilla. Tämä siksi, että muut indikaattorit (FNN, AC ja Shannon) eivät osoita näissä aikasarjoissa mitään erikoisempaa rakennetta muutamaa poikkeusta lukuunottamatta (liite 1.).

Suurimpana ongelmana projektin aikana oli käytettävän datan vähäinen määrä (katso kuva 20). Kirjallisuudessa ja muualla vastaavan kaltaisissa analyyseissä *lyhyt* aikasarja on ollut tuhat datapistettä, ja yleensä analyysejä on tehty aikasarjoille, joiden pituus on ollut esim. 5000 datapistettä. Pitkät näytteet historiadatasta parantavat menetelmien mahdollisuuksia löytää kaikki oleelliset jaksolliset muutokset datasta, vähentävät yksittäisten satunnaisten "ylilyöntien" ja muiden epänormaalien tapahtumien vaikutusta lopulliseen analyysiin. Koko komponenttikannassa ei ollut kuin muutama komponentti, jotka pääsivät edes lähelle datapisteiden alarajaa, joka pyrittiin määrittämään esimerkkitapauksia hyväksi käyttäen (400 datapistettä). Lisäksi matalan kaaoksen aikasarjojen analysointiin tarvitaan vähemmän dataa kuin korkeasti kaoottisen analysointiin; jos dataa on liian vähän aikasarja voidaan tulkita ainoastaan satunnaiseksi (kuten tässä tutkimuksessa on käynyt).

Yksi tärkeä tekijä aikasarja-analyyseissä on jaksot: tässä tapauksessa dominoivana on kohtuullisen lyhyt, viikon pituinen jakso. Tämä johtuu siitä, että tehtaantuo- tuotanto on yleensä kerran viikossa seisahtunut (sunnuntaisin). Toinen, ehkä vieläkin olennaisempi jakso on vuosi. Kaikki vuosittaiset katkokset, kuten kesä- ja joululomat, yleensä hiljaisempi kesäaika jne. ovat oleellisia analysoinnin kannalta. Jotta tämä vuoden

pituinen jakso saataisiin mukaan analyysiin, täytyisi dataa olla vähintään kahden vuoden, ellei vieläkin pidemmältä ajalta. Vaaditun ennusteajan puitteissa ehtii tapahtua jo melkoisia muutoksia, jotka ovat sidonnaisia juuri tähän vuosivaihteluun.

Viikon mittainen jakso aiheuttaa dataan voimakasta vaihtelua. Tämän vaikutusta on pyritty vähentämään poistamalla datasta kaikki nk. 0-päivät. Eli käytännössä siirrytään normaalista ajan mittaamisesta komponenttikohtaiseen kalenteriin, joka on lähempänä tehdaskalenteria. Viikon jakso aiheuttaa vahvan "kohinan" dataan; esim. maanantai ja perjantai/lauantai voivat olla päiviä, jolloin tuotantoa (eli komponentin kulutusta) ei ole niin paljon kuin muina viikoppäivinä. Analysointia voisi helpottaa, jos päivätason käsittelystä siirryttäisiin viikkotasolle. Näin merkittävä osa voimakkaita, epäoleellisia vaihteluita saataisiin hävitettyä datasta ja sen analysointi voisi olla helpompaa. Vastaavasti datan vähäinen määrä on viikkotason analyysissä vieläkin pahempi ongelma: dataa pitäisi olla ainakin $400 \times 7 = 2800$ päivältä, mikä on noin 8 vuotta!

Vaaditut pitkät historiadatan määrät tuovat esiin toisen perustavaa laatua olevan ongelman. Ei ole lainkaan selvää, että komponentit ovat olleet olemassa tai säilyvät tuotannossa nopeasti muuttuvalla elektroniikkateollisuuden alalla tarpeeksi pitkiä aikoja (3-8 vuotta). Tietoliikenneverkkoihin ja tietotekniikkaan liittyvä teknologia on muuttunut aika tavalla viimeisen kahdeksan vuoden aikana.

Yksi vaihtoehto tutkittaessa komponenttien kulutusta on tuoda lisää informaatiota analysoimalla yksittäisten komponenttien välisiä suhteita kulutuksessa. Esimerkiksi, jos komponenttia A kuluu n kappaletta, niin yleensä komponenttia B kuluu $5 \times n$ kappaletta. Tällaisten ristikorrelaatioiden etsiminen vaatii yksityiskohtaista tietoa tuotannosta ja tuotteista, sillä tilastollinen haku on erittäin raskasta, esim. parikorrelaatiot vaativat $N \times (N-1)/2$ hakua, 8322 komponentin tapauksessa analysoitavia korrelaatioita olisi 34 623 681 kappaletta! Korkeamman asteen korrelaatioiden (monia eri komponenttia) analysointi on jo ylivoimaisen raskasta; kolmen komponentin korrelaation tapauksessa tutkittavien korrelaatioita määrä on $N \times (N-1) \times (N-2)/3$, eli 8322 komponentin tapauksessa 192 046 017 280!

Lähdeluettelo

- Abarbanel, Henry D. I., Brown, Reggie, Sidorowich, John J., Tsimring, Lev Sh. 1993, *Rev. Mod. Phys.* 65 (4) 1331-1392
- Eckmann, J.-P., Ruelle, D. 1985, *Rev. Mod. Phys.* 57 617-656
- Farmer, J. Doyné 1980, *Physica D* 4 366-393
- Fausett, Laurene 1994, *Fundamentals of Neural Networks, Architectures, Algorithms, and Applications*, Prentice-Hall
- Grassberger, P., Procaccia, Itamar, *Phys. Rev. Letters* 50 346-349
- Greenside, Henry S. 1996, *preprint chao-dyn/9612004*
- Hilborn, Robert C. 1994, *Chaos and Nonlinear Dynamics*, Oxford University Press
- Kaspar, F., Schuster, H. G. 1987, *Phys. Rev. A* 36, 842
- Schwarz, U., Benz, A. O., Kurths, J., Witt, A. 1993, *Astronomy and Astrophysics* 277 215-224
- Wan, E. 1990, *International Joint Conference on Neural Networks* (San Diego 1990) Vol. 1, 575-580
- Wan, E. 1993, *Proceedings of the NATO Advanced Research Workshop on Time Series Prediction and Analysis*, (Santa Fe, New Mexico, May 14-18 1992), A. Weigend and N. Gershenfeld, eds. Addison-Wesley

Liite 1. Komponenttlista

Lista komponenteista, jotka sisältävät *average mutual information* kuvaajan mukaan informaatiota. Komponentit on seulottu kappaleessa 7 esitettyjen kriteerien mukaisesti. ID-nro on komponentin järjestysnumero datatiedostossa, datapisteitä on nolasta poikkeavien datapisteiden (laskennassa käytettyjen) lukumäärä, alkupäivämäärä on se päiväys, jolla on ensimmäinen datapiste. Kesto tarkoittaa ensimmäisen ja viimeisen datapisteen välistä aikaeroa päivinä. Shannon-sarakkeessa on komponentille laskettu *Shannon-informaatio* ja AC-sarakkeessa *algorithmic complexity*.

Komponentit, joilla *Shannon-informaation* ja *algorithmic complexityn* välinen erotus on kappaleen 5 lopulla esitetyn kriteerin mukaisesti vähemmän kuin 0.1, on alleviivattu. Eli käytännössä paras mahdollinen tapaus on sellainen, jolla on eo. erotus mahdollisimman pieni, varsinaiset *Shannon-informaation* ja *algorithmic complexityn* arvot mahdollisimman pieniä ja mahdollisimman paljon datapisteitä.

HUOM! Tässä yhteydessä kannattaa muistaa, että kaikilla, myös tässä listassa olevilla komponenteilla, sisäisen rakenteen tärkein indikaattori (FNN) antoi negatiivisen tuloksen! Lisäksi testitapauksilla vaadittiin vähintään 400 datapistettä luotettavien tulosten saamiseksi.

ID-nro	Koodi	Datapisteitä	Alkupvm.	Kesto	Shannon	AC
46	E00180000-0	185	5.1.1996	419	0.585	0.692
392	E09618049-A	165	3.1.1996	706	0.683	0.982
399	E09619194-0	118	9.1.1996	679	0.620	0.817
467	E09626664-0	206	3.1.1996	708	0.429	0.672
525	E09628471-A	273	3.1.1996	710	0.550	0.682
528	E09628499-0	190	4.1.1996	708	0.741	0.996
546	E09628657-0	173	8.1.1996	705	0.750	0.903
562	E09628767-0	173	9.1.1996	704	0.845	1.074
691	E10008974-B	167	6.3.1996	632	0.653	0.840
693	E10009009-0	195	2.1.1996	709	0.911	1.248
694	E10009010-0	189	3.1.1996	708	0.916	1.160
711	E10009486-A	197	3.1.1996	709	0.736	0.967
1080	F09977451-A	137	3.1.1996	708	0.030	0.207
1152	F10000064-G	272	1.7.1996	528	0.919	1.041
1164	F10000109-A	117	1.7.1996	521	0.873	1.292
1229	F10000276-B	113	14.10.1996	421	0.730	0.966
1683	K01412046	343	9.4.1996	612	0.674	0.638
1784	K01420000	446	1.7.1996	446	0.363	0.509
1821	K01420080	341	4.4.1996	617	0.525	0.592
1850	K01420212	185	3.7.1996	527	0.414	0.489
1930	K01420730	227	1.7.1996	529	0.678	0.828
1958	K01420956	170	1.7.1996	444	0.770	1.046
1975	K01421090	169	1.7.1996	444	0.772	1.051
1976	K01421092	169	1.7.1996	444	0.776	1.051
1978	K01421098	170	1.7.1996	444	0.770	1.046
1997	K01421842	174	1.7.1996	520	0.774	1.069
2017	K01422918	169	1.7.1996	444	0.755	1.007

ID-nro	Koodi	Datapisteitä	Alkupvm.	Kesto	Shannon	AC
2018	K01422984	332	1.7.1996	528	0.784	0.933
2383	K02300614	177	3.1.1996	710	0.808	1.013
2478	K02310495	136	3.1.1996	708	0.030	0.209
2506	K02311238	101	16.7.1996	515	0.7625	0.989
2509	K02311246	136	3.1.1996	708	0.030	0.209
2600	K02400724	186	2.1.1996	709	0.757	0.932
2659	K02406045	167	6.2.1996	676	0.960	1.326
2730	K02504850	173	2.1.1996	711	0.879	1.160
2738	K02505782	176	4.1.1996	709	0.811	1.017
2742	K02508444	176	2.1.1996	711	0.813	1.017
2743	K02509030	177	2.1.1996	711	0.808	1.013
2801	K03204368	267	2.1.1996	711	0.802	0.966
2819	K03602504	184	2.1.1996	709	1.002	1.308
2820	K03602506	199	2.1.1996	711	0.877	0.998
2848	K03605380	224	2.1.1996	711	0.912	1.046
2861	K03505866	211	3.1.1996	710	0.975	1.171
2885	K03606985	213	2.1.1996	711	1.006	1.198
2888	K03607114	134	2.4.1996	620	0.970	1.213
2905	K03607820	170	1.10.1996	437	0.855	1.046
3113	K04101779	441	3.1.1996	710	1.033	1.096
3135	K04103410	213	2.1.1996	711	0.972	1.126
3140	K04104652	117	16.7.1996	515	0.670	1.057
3195	K04110075	446	2.4.1996	620	1.062	1.125
3200	K04110088	344	9.4.1996	612	0.781	0.857
3220	K04110212	378	1.7.1996	529	0.925	0.974
3227	K04110269	420	4.4.1996	618	1.014	1.079
3351	K04207620	215	2.1.1996	711	0.961	1.189
3359	K04210082	136	3.1.1996	708	0.030	0.209
3510	K04308967	126	4.1.1996	708	0.639	0.831
3641	K04320097	157	15.1.1996	694	0.855	1.069
4000	K04341709	442	29.3.1996	624	0.816	0.875
4008	K04341812	416	4.4.1996	618	0.708	0.816
4255	K04345584	326	1.7.1996	529	1.000	1.127
4375	K04348748	308	1.7.1996	529	0.930	1.020
4441	K04349640	309	1.7.1996	529	0.965	1.044
4457	K04350750	139	11.1.1996	699	0.823	1.178
4667	K04370179	132	8.1.1996	705	0.668	0.854
4710	K04370660	306	1.7.1996	529	0.970	1.106
4946	K05114662	174	1.7.1996	520	0.771	0.984
4995	K05203292	156	3.1.1996	708	0.815	1.074
5042	K05301226	376	10.4.1996	611	0.804	0.865
5083	K05401136	160	2.1.1996	709	0.802	1.053
5094	K05402433	285	3.1.1996	707	0.370	0.458
5136	K05407054	262	9.1.1996	704	0.832	1.012
5194	K05414728	322	8.2.1996	672	0.825	0.931
5205	K05414796	168	1.7.1996	444	0.778	1.056
5248	K05416997	228	5.1.1996	708	0.925	1.134
5271	K05418108	117	16.7.1996	515	0.695	0.998
5296	K05419055	168	1.7.1996	444	0.774	1.056
5311	K05420326	118	16.7.1996	515	0.795	1.108
5380	K05430490	130	15.7.1996	516	0.659	1.026

ID-nro	Koodi	Datapisteitä	Alkupvm.	Kesto	Shannon	AC
5520	K05440430	178	4.1.1996	709	0.845	1.050
5687	K05447868	239	2.1.1996	505	0.610	0.827
5800	K05461018	178	29.3.1996	623	0.886	1.092
5861	K05463096	101	17.7.1996	510	0.772	1.121
5910	K06112044	211	8.1.1996	645	0.683	0.951
6075	K06403480	342	2.1.1996	711	0.849	0.935
6199	K07100602	181	8.1.1996	698	0.767	1.036
6471	K07334240	106	4.1.1996	342	0.469	0.571
6505	M09006800-A	163	2.1.1996	640	0.448	0.541
6588	M09265499-A	159	10.1.1996	701	0.859	1.012
6692	M09284469-0	186	5.1.1996	708	0.863	1.094
7044	M09601888-0	144	16.1.1996	695	0.959	1.295
7056	M09602641-E	147	19.1.1996	692	0.836	1.078
7059	M09602715-C	214	3.1.1996	707	0.707	0.868
7081	M09604582-0	143	4.1.1996	707	0.887	1.152
7083	M09605064-D	140	17.1.1996	694	0.750	1.018
7088	M09605275-B	129	18.1.1996	407	0.393	0.707
7089	M09605385-A	281	3.1.1996	710	0.337	0.492
7090	M09605794-0	156	15.1.1996	696	0.849	1.027
7102	M09612537-A	160	3.1.1996	496	0.396	0.641
7148	M09619451-0	147	10.1.1996	701	0.801	1.029
7207	M09622360-0	200	5.1.1999	691	0.537	0.688
7242	M09624396-A	281	5.1.1996	641	0.527	0.753
7253	M09624772-0	142	11.1.1996	700	0.735	0.957
7305	M09626531-0	159	3.1.1996	704	0.553	0.644
7336	M09626994-A	237	3.1.1996	457	0.903	1.132
7365	M09627730-0	296	12.1.1996	701	0.653	0.666
7386	M09628031-A	229	5.1.1996	707	0.760	0.959
7393	M09628061-0	198	3.1.1996	708	0.601	0.771
7397	M09628076-0	119	24.1.1996	687	0.740	0.985
7398	M09628085-0	145	11.1.1996	373	0.586	0.941
7421	M09628221-0	234	8.1.1996	705	0.623	0.774
7441	M09628312-A	298	3.1.1996	624	0.807	0.938
7459	M09628391-B	211	5.1.1996	707	0.534	0.769
7478	M09628502-D	195	10.1.1996	701	0.737	0.819
7480	M09628522-0	197	8.1.1996	703	0.736	0.929
7481	M09628529-0	277	2.1.1996	625	0.665	0.732
7488	M09628559-0	170	4.1.1996	709	0.539	0.828
7501	M09628627-A	160	5.1.1996	706	0.806	1.053
7505	M09628643-0	162	2.1.1996	423	0.454	0.725
7525	M09628708-B	164	16.1.1996	626	0.469	0.763
7553	M09628807-0	139	8.1.1996	705	0.095	0.256
7575	M09628911-A	220	2.1.1996	710	0.549	0.814
7595	M09628989-A	352	2.1.1996	711	0.847	0.913
7775	M10001101-A	198	5.1.1996	706	0.716	0.925
7826	M10001297-0	177	4.1.1996	709	0.943	1.181
7842	M10001353-A	102	10.7.1996	520	0.793	1.308
7843	M10001354-A	102	10.7.1996	520	0.793	1.308
7904	M10001641-0	134	29.1.1996	683	0.666	0.791
7912	M10001678-0	141	9.1.1996	501	0.788	1.063

ISBN 951-42-5873-8

ISSN 1238-9404

Oulun yliopisto

Säätötekniikan laboratorio - Sarja B

Toimittaja: Leena Yliniemi

1. Yliniemi L & Koskinen J, Rumpukuivaimen sumea säätö. Joulukuu 1995. 17 s. 6 liitettä. ISBN 951-42-4301-3.
2. Leiviskä K, Rauma T, Ahola T, Juuso E, Myllyneva J & Alahuhta P, Sumea mallintaminen, viritys ja säätö. Tammikuu 1996. 44 s. 951-42-4348-X.
3. Altavilla M, Koskinen J & Yliniemi L, Rumpukuivaimen säätö neuroverkolla. Tammikuu 1996. 12 s. ISBN 951-42-4373-0.
4. Myllyneva J, Leiviskä K, Heikkinen M, Kortelainen J & Komulainen K, Sumean säädön käyttömahdollisuudet hiertämön ohjauksessa. Huhtikuu 1997. 52 s. ISBN 951-42-4647-0.
5. Leiviskä K & Heikkinen M, TMP-prosessin mallintaminen ja mallipohjainen säätö. Huhtikuu 1997. 68 s. ISBN 951-42-4646-2.
6. Jaako J, Nopeusyhtälön parametrien sovittaminen. Huhtikuu 1998. 25 s. ISBN 951-42-4961-5.
7. Myllyneva J, Kortelainen J, Latva-Käyrä K, Nystedt H & Leiviskä K, Hiertämön laatusaadöt. Syyskuu 1998. ISBN 951-42-5023-0.
8. Lähteenmäki M & Leiviskä K, Tilastollinen prosessinohjaus: perusteet ja menetelmät. Lokakuu 1998. ISBN 951-42-5064-8.
9. Tervahartiala P & Leiviskä K, Tilastollinen prosessinohjaus: ohjelmistoverailu. Elokuu 1999. ISBN 951-42-5343-4.
10. Jaako J, Eräitä optimointitehtäviä. Syyskuu 1999. 39 s. ISBN 951-42-5352-3.
11. Jaako J, Yksinkertaisia prosessimalleja. Syyskuu 1999. 73 s. ISBN 951-42-5353-1.
12. Jaako J, MATLAB-ohjelman käyttö eräissä prosessiteknisissä laskuissa. Syyskuu 1999. 61 s. ISBN 951-42-5354-X.
13. Jaako J, Säätötekniikan laboratorion opetuskokeiluja I - Portfoliomuotoisen kurssin toteutus ja tulokset. Helmikuu 2000. 28 s. ISBN 951-42-5544-5.
14. Ahola T, Ruuska J, Juuso E & Leiviskä K, Paperikoneen katkoherkkyysindikaattori. Helmikuu 2000. 33 s. ISBN 951-42-5563-1.
15. Ylikunnari J, InTouch valvomo-ohjelmiston implementointi lämmönsiirron identifiointiprosessiin (PS II:n harjoitustyölaiteisto). Maaliskuu 2000. ISBN 951-42-5568-2.
16. Mäki T & Juuso E, Tapahtumapohjainen sumea lingvistinen yhtälöjärjestelmä lääkevalmisteiden koostumusten ja valmistusprosessien tutkimuksessa. Kesäkuu 2000. ISBN 951-42-5678-6.
17. Jaako J, Säätötekniikan laboratorion opetuskokeiluja II - Apuopettaja opettajan apuna. Elokuu 2000. 22 s. ISBN 951-42-5742-1.
18. Sivonen J, Johdatus säätötekniikkaan, opetuslaitteiston suunnittelu ja toteutus. Syyskuu 2000. 20 s. ISBN 951-42-5795-2.
19. Mutka P., Neuraalilaskenta ja epälineaarinen dynamiikka komponenttien kulutus- ja myynti-ennusteiden laatimisessa. 49 s. ISBN 951-42-5773-8.

Oulun yliopistopaino

Oulu 2000