TARTU ÜLIKOOL
MATEMAATIKA-INFORMAATIKA TEADUSKOND
Matemaatilise statistika instituut

Cliona Georgia Dalberg

Eesti elektritarbimise prognoos

Magistritöö
finants- ja kindlustusmatemaatika erialal (30 EAP)

Juhendajad: Joosep Lassmann, MSc, Eesti Energia
Raul Kangro, PhD, Tartu Ülikool

Tartu 2015

Eesti elektritarbimise prognoos

Käesolevas magistritöös prognoositakse Eesti elektritarbimist 24 tundi ette. Antakse ülevaade tugivektorregressiooni teooriast ning kasutatavast paketist R tarkvaras. Koostatakse ennustamiseks lineaarse regressiooni mudelid ning tehakse nende analoogid tugivektorregressiooni abil. Võrdlemiseks kasutatakse ka ARIMA mudelit. Tulemusi hinnatakse 2015. aasta jaanuari ning veebruari prognooside keskmise suhtelise vea ning keskmise ruutvea põhjal. Mudeleid parandatakse argumenttunnuste lisamise ning muutmisega. Lõplik valik parima mudeli osas tehakse uue testperioodi kaasamisel.

Märksõnad: elektritarbimine, prognostika, regressioonanalüüs, aegridade analüüs, tehisõpe

Estonian electric load forecast

This thesis concentrates on forecasting Estonian electric load 24 hours ahead. The overview of support vector regression and available packages in R software are given. Linear models are built and transformed to support vector regression models. Also one time series model is considered for comparison. Performances of models are evaluated by mean absolute percentage error and root mean square error of January and February 2015. Models are modified by adding different variables. Final selection is based on a new test period.

Key words: electricity consumption, prediction, regression analysis, analysis of time series, automatic learning

Sisukord

1 Sissejuhatus

Elektritarbimise võimalikult täpne ennustamine on ettevõtte jaoks väga tähtis. Nord Pool Spoti andmetel maksis Eestis aastal 2013 1MW/h elektrit keskmiselt 43,14 eurot ning aastal 2014 37,61 eurot. Keskmine tarbimine tunnis oli 2013. aastal 905,0751 MW. Neljaprotsendise eksimuse juures oleks võinud aastal 2013 kahju ulatuda ühes tunnis üle 1550 euro, 2014. aastal üle 1350 euro.

Selge on see, et elektritarbimine on juhuslik. Juhuslike suuruste prognoosimiseks kasutatakse mitmesuguseid mudeleid, mille hulgas on hästituntud lineaarse regressiooni mudel, aegridade jaoks mõeldud ARIMA mudelid ning suhteliselt hiljuti väljatöötatud tugivektorregressiooni mudelid. Töö eesmärgiks on kindlaks teha, kas tugivektorregressiooniga õnnestub saada elektritarbimisele paremaid lühiajalisi prognoose, kui kasutusel olevad ARIMA mudelid saavad.

Magistritöö koosneb kolmest osast. Esimene osa annab põgusa ülevaate tugivektorregressiooni teooriast ning rakendamise võimalusest tarkavaras R. Teoorias tutvustatakse kõigepealt lineaarset juhtu ning seejärel üleminekut mittelineaarsele juhule koos tüüpilisemate tuumafunktsioonidega. Rakendamise puhul antakse ülevaade tugivektoritega töötamise paketist *Caret* ning räägitakse skaleerimisest ning parameetritele analüütiliste lähendite leidmisest.

Teine ja kolmas osa puudutavad praktilist käsitlust. Esmalt sobitatakse andmetele lineaarseid mudeleid, mis aitavad leida parimaid tarbimist iseloomustavaid argumenttunnuseid ning mõista, kuidas kasutada prognoosimisel tarbimise ajalugu. Lineaarste mudelite sobitamisega luuakse võrdlusbaas, mis aitab tugivektorregressiooniga saavutatud tulemusi hinnata. Lineaarse regressiooni kõrval katsetatakse ka aegridade analüüsi vahendeid ning tehakse üks ARIMA mudel. Saadud mudelite puhul tuuakse välja tugevad ja nõrgad küljed ning hinnatakse nende prognoosivõimet. Lisaks illustreeritakse tulemusi graafiliselt. Seejärel keskendutakse tugivektorregressiooni mudelite sobitamisele.

Kolmandas osas täpsustatakse parimaks osutunud mudeleid, lisades tarbimise ajalugu iseloomustavaid tunnuseid. Koostatakse koondtabelid ning valitakse välja parim mudel. Lisas on tarkvara R koodid. Tööle on kaasa pandud ka failid koos koodide, andmestike ja tulemustega.

Autor soovib tänada juhendajaid Raul Kangrot ning Joosep Lassmanni suunamise, märkuste ning paranduste eest.

2 Ülevaade tugivektorregressiooni teooriast ja rakendamisest

2.1 Tugivektorregressiooni idee

Tugivektorid on masinõppes olulisel kohal ning neid kasutatakse enamasti klassifitseerimisprobleemide lahendamisel, kus objekti klassikuulumise otsus võetakse vastu väikese hulga treeningandmete (tugivektorite) põhjal- omadus, mida nimetatakse hõreduseks. Hõredus tuleneb sellest, et klassifitseerimisülesande korral on tähtis otsustada, kus paiknevad klassidevahelised piirid argumenttunnuste ruumis ning piiri määravad vaid osad vaatlused, mis on piirilähedased ja neid rikuvad. Selgelt õigetesse piiridesse kuuluvate vaatluste asukohad piirkonna sees ei ole olulised ja selliste vaatluste argumentide väikesed muudatused ei mõjuta piiride asukohta. Sarnane piirkond peab olema ka regressioonhinnangu korral [13, lk 252]. Hõreduse ülekandmiseks reaalarvuliste väärtustega juhuslike suuruste lähendamisele, on kasutusele võetud epsilon-tundetu kaofunktsioon [13, lk 251]:

$$|y - f(x)|_{\varepsilon} = max\{0, |y - f(x)| - \varepsilon\}. \tag{1}$$

Empiiriliseks riskiks nimetatakse suurust

$$R(f) = \frac{1}{l} \sum_{i=1}^{l} L(f(x), y), \text{ kus } L(f(x), y) \text{ on kaofunktsioon.}$$

Piisavalt suure ε korral on võimalik, et empiiriline risk saab epsilon-tundetu kaofunktsiooni korral väärtuseks nulli, mis ei tähenda aga, et $f(x)$ oleks sobiv ennustamiseks. Seega jõutakse ruutplaneerimise ülesandeni.

Järgnev ülevaade põhineb artiklil [15]. Antud on treeningandmed $\{(x_1, y_1), ..., (x_l, y_l)\} \subset \mathcal{X} \times \mathbb{R}$, kus $\mathcal{X}$ tähistab tunnusvektorite ruumi. Epsilon-tugivektorregressiooni peamine eesmärk on sobitada selline funktsioon, mille korral kaofunktsiooni 1 abil leitud kadu oleks treeningandmete korral nii väike kui võimalik. Funktsioon ise peaks olema samal ajal argumentide muutumise korral võimalikut aeglaselt muutuv. Teisisõnu, epsilonist väiksemad vead ei paku meile huvi (kuna need ei mõjuta tulemust), kuid suuremate vigade poolt põhjustatud kadu peab olema minimaalne.

2.2 Lineaarne juht

Tugivektorregressiooni tavajuht kasutab argumenttunnuste suhtes lineaarseid funktsioone. Lineaarse funktsiooni võib kirja panna kujul

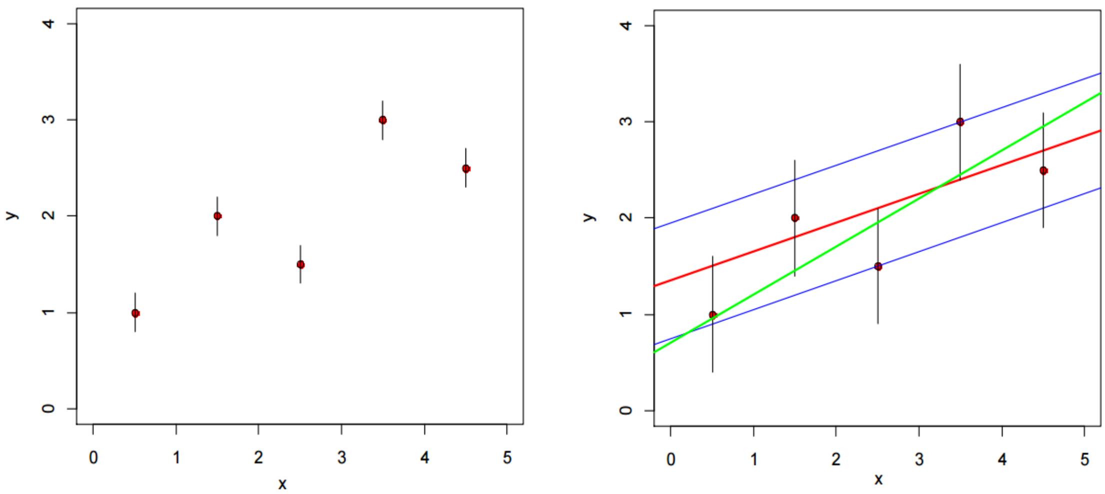
$$f(x) = \langle w, x \rangle + b \qquad w,x \in \mathcal{X},\ b \in \mathbb{R}, \tag{2}$$

kus $\langle \cdot, \cdot \rangle$ tähistab skalaarkorrutist ruumis $\mathcal{X}$. Selle funktsiooni jaoks tähendab väike tõus võimalikult väikese normiga w väärtust.

Üks variant aeglase muutumise tagamiseks on minimiseerida norm $\|w\|^2 = \langle w, w \rangle$. Seega saab konstrueerida kumeraoptimiseerimise probleemi:

$$\begin{aligned} &\text{minimiseerida } \frac{1}{2}\|w^2\| \\ &\text{tingimusel, et } |y_i - f(x_i)| \le \varepsilon,\ i = 1, 2 \ldots, l \end{aligned} \tag{3}$$

Hulka, mis sisaldab selliseid vektoreid w ning vabaliikmeid b, mille korral kõik kitsendused on täidetud, nimetatakse lubatavaks hulgaks. Eespool toodud lähenemine on piisav, kui etteantud täpsuse epsilon korral on lubatav hulk mittetühi. Mittetühja lubatava hulga saaks tagada, valides väga suure epsiloni, kuid see viiks ebatäpsete tulemusteni. Lihtsa illustreeritud näite võib leida Mušnikovi magistritööst [12, lk 11-13], kus näidatakse, et epsiloni 0,2 korral (joonisel 1 vasakul) ei ole võimalik leida sellist lineaarset funktsiooni, mis rahuldaks toodud kitsendusi. Suurendades epsiloni 0,6-ni (paremal) tekib mittetühi lubatav hulk ning leidub mitmeid lahendiks sobivaid variante. Joonisel on kaks võimalikku lahendit, väikseima võimaliku tõusuga lahend on toodud punasega.



Joonis 1: Vasakul mittelahenduv probleem $\varepsilon = 0,2$ korral [12, lk 12] ning paremal näited lahenditest lõdvendatud kitsenduse $\varepsilon = 0,6$ korral [12, lk 13]

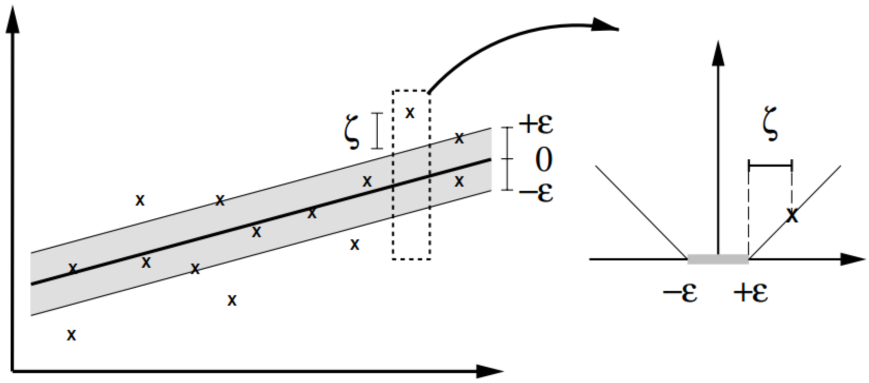
Lubatava hulga olemasoluks lõdvendatakse kitsendusi, tuues sisse positiivseid ning negatiivseid absoluutväärtuselt epsilonist suuremaid prognoosivigasid iseloomustavad muutujad ξ ja ξ^* ning lahendatakse ülesanne

$$\text{minimiseerida } \frac{1}{2}\|w^2\| + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \tag{4}$$

$$\text{tingimusel, et } \begin{cases} y_i - \langle w, x_i\rangle - b \leq \varepsilon + \xi_i \\ \langle w, x_i\rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \qquad\qquad \xi_i, \xi_i^* \quad \geq 0 \end{cases}$$

Konstant C (regularisatsiooniparameeter) määrab tasakaalu vektori w tõusu ning epsilon-tundetu kaofunktsiooni abil arvutatud kao vahel [9]. C annab suurema või väiksema kaalu nendele absoluutväärtuselt epsilonist suurematele prognoosivigasid iseloomustavatele muutujatele, mis mõjutavad mudeli keerukust ning seeläbi stabiilsust ja ülesobitamist [14].

Joonis 2 iseloomustab epsilon-tundetut piirkonda ning suuremaid prognoosivigu iseloomustavaid tunuseid. Ainult need punktid, mis asuvad väljaspool varjutatud ala tekitavad kahju. Kahju tekib lineaarselt- mida suurem on eksimus, seda rohkem kahju tekib.



Joonis 2: Ühemõõtmelise lineaarse regressiooni juhu epsilontundetu piirkond[15, lk 2]

Enamikul juhtudest lahendatakse ülesanne 4 sellega duaalse ülesande lahendamise abil. Probleem lahendatakse parameetri C erinevate väärtuste korral ja pärast valitakse valideerimise abil optimaalseim väärtus[3].

Peamine idee on konstrueerida sihifunktsioonist Lagrange'i funktsioon ning vastavad kitsendused, kasutades duaalseid muutujaid. On võimalik näidata, et nii esialgse kui duaalse ülesande lahend on vastava ülesande sihifunktsiooni sadulpunktiks [13, lk 254-255].

Üldise juhu kohta on teoreetilise meetodi üksikasjalikult läbi teinud oma magistritöös Mušnikov [12, lk 10-11], võttes aluseks ruutplaneerimise probleemi (3):

$$\begin{aligned}&\text{minimiseerida } \frac{1}{2}\|w^2\|\\&\text{kitsendusega } |y_i - f(x_i)| \leq \varepsilon\end{aligned}$$

Mušnikov on kirjeldanud järgmised sammud:

1. Lagrange'i funktsiooni moodustamine

$$L = \frac{1}{2}\|w\|^2 - C\sum_{i=1}^{l} \alpha_i(\varepsilon + y_i - f(x_i)) - \sum_{i=1}^{l} \alpha_i^*(\varepsilon - y_i + f(x_i))$$
$$\text{tingimusel } \alpha_i, \alpha_i^* \geq 0, i = 1, \ldots, l$$

2. Lineaarse funktsiooni 2 asendamine Lagrange'i funktsiooni
3. Slateri tingimustele viitamine (kui Slateri tingimused on rahuldatud, siis on funktsioonil lahendi kohal sadulpunkt nii esialgsete kui duaalsete muutujate suhtes)[13, ptk 6]
4. Kuhn-Tuckeri teoreemi kasutamine (ruutplanerimise probleemi lahendi leidmine on samaväärne Lagrange'i funktsiooni sadulpunktide leidmisega)[13, lk 166]
5. Lagrange'i funktsiooni osatuletise võtmine ning nulliga võrdustamine ja asendamine punktis kaks saadud võrdusesse
6. Duaalse ülesandeni jõudmine

Tuletame sama skeemi kohaselt ülesande 4 duaalse ülesande. Sellel juhul on Lagrange'i funktsiooniks:

$$\begin{aligned} L := \; & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) - \sum_{i=1}^{l}(\eta_i\xi_i + \eta_i^*\xi_i^*) \\ & - \sum_{i=1}^{l}\alpha_i(\varepsilon + \xi_i - y_i + \langle w, x_i\rangle + b) \\ & - \sum_{i=1}^{l}\alpha_i^*(\varepsilon + \xi_i^* + y_i - \langle w, x_i\rangle - b) \end{aligned} \tag{5}$$

Siin $\eta_i, \eta_i^*, \alpha_i, \alpha_i^*$ on Lagrange'i kordajad, seega duaalsed muutujad peavad rahuldama kitsendusi

$$\alpha_i^{(*)}\ ,\ \eta_i^{(*)} \geq 0.$$

Tähistus $^{(*)}$ viitab korraga nii tärniga kui tärnita muutujale. Sadulpunkti tingimusest järeldub, et Lagrange'i funktsiooni osatuletised esialgsete muutujate järgi peavad olema nullid.

$$\partial_b L = \sum_{i=1}^{l}(\alpha_i^* - \alpha_i) = 0 \tag{6}$$

$$\partial_w L = w - \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)x_i = 0 \tag{7}$$

$$\partial_{\xi_i^{(*)}} L = C - \alpha_i^{(*)} - \eta_i^{(*)} = 0 \tag{8}$$

Võrreldes Mušnikovi lahendatud probleemiga on lisandunud tingimus ehk osatuletised $\xi^{(*)}$ järgi. Duaalse ülesande saamiseks tuleb asendada 6, 7 ja 8 Lagrange'i funktsiooni 5 :

$$\begin{aligned}
L \overset{5}{=} & \frac{1}{2}\Big\langle \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)x_i, \sum_{j=1}^{l}(\alpha_j - \alpha_j^*)x_j\Big\rangle + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) - \sum_{i=1}^{l}(\eta_i\xi_i + \eta_i^*\xi_i^*) \\
& - \sum_{i=1}^{l}\alpha_i(\varepsilon + \xi_i - y_i + \Big\langle \sum_{j=1}^{l}(\alpha_j - \alpha_j^*)x_j, x_i\rangle + b) \\
& - \sum_{i=1}^{l}\alpha_i^*(\varepsilon + \xi_i^* + y_i - \langle\sum_{j=1}^{l}(\alpha_j - \alpha_j^*)x_j, x_i\rangle - b) = \\
= & \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\langle x_i x_j\rangle \\
& + \underbrace{C\sum_{i=1}^{l}(\xi_i + \xi_i^*) - \sum_{i=1}^{l}(\eta_i\xi_i + \eta_i^*\xi_i^*)\sum_{i=1}^{l}\alpha_i\xi_i - \sum_{i=1}^{l}\alpha_i^*\xi_i^*}_{(\sum_{i=1}^{l}(C-\eta_i-\alpha_i)\xi_i - \sum_{i=1}^{l}(C-\eta_i^*-\alpha_i^*)\xi_i^* = 0\ (8)} \\
& \underbrace{-\sum_{i=1}^{l}\alpha_i\varepsilon - \sum_{i=1}^{l}\alpha_i^*\varepsilon}_{\varepsilon\sum_{i=1}^{l}(\alpha_i+\alpha_i^*)} + \underbrace{-\sum_{i=1}^{l}\alpha_i y_i - \sum_{i=1}^{l}\alpha_i^* y_i^*}_{y_i\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)} \underbrace{-\sum_{i=1}^{l}\alpha_i b + \sum_{i=1}^{l}\alpha_i^* b}_{b\sum_{i=1}^{l}(\alpha_i^*-\alpha_i)=0\ (6)} \\
& \underbrace{-\sum_{i=1}^{l}\alpha_i\langle\sum_{j=1}^{l}(\alpha_j - \alpha_j^*)x_j, x_i\rangle + \sum_{i=1}^{l}\alpha_i^*\langle\sum_{j=1}^{l}(\alpha_j - \alpha_j^*)x_j, x_i\rangle}_{\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_j-\alpha_j^*)\big\langle x_j, x_i\big\rangle(\alpha_i^*-\alpha_i)=0\ (7)}
\end{aligned}$$

Seega optimiseerimisülesandeks on

$$\text{maksimiseerida} \begin{cases} -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\langle x_i, x_j\rangle \\ -\varepsilon\sum_{j=1}^{l}(\alpha_i + \alpha_i^*) + \sum_{j=1}^{l} y_i(\alpha_i - \alpha_i^*) \end{cases} \tag{9}$$

$$\text{tingimusel} \qquad \sum_{j=1}^{l}(\alpha_i + \alpha_i^*) = 0 \text{ ja } \alpha_i, \alpha_i^* \in [0, C]$$

Tingimus $\alpha_i^{(*)} \in [0, C]$ tuleb seosest 8: $\eta_i^{(*)}$ saab avaldada kujul $\eta_i^{(*)} = C - \alpha_i^{(*)}$ ning $\eta_i^{(*)}$ peavad olema suuremad nullist.
Seosest 7 saab avaldada w tunnusvektorite lineaarse kombinatsioonina:

$$w = \sum_{j=1}^{l}(\alpha_i + \alpha_i^*)x_i$$

ning $f(x)$ saab 2 järgi kuju: $f(x) = \sum_{j=1}^{l}(\alpha_i + \alpha_i^*)\langle x_i, x\rangle + b$. Seega ei sõltu mudeli keerukus ruumi $\mathcal{X}$ mõõtmetest, vaid ainult tugivektorite arvust. Tasub tähele panna, et kogu algoritmi vältel kasutatakse vaid andmete skalaarkorrutisi. Lahendades optimiseerimisülesande 9 saadakse α_i, α_i^* ning x_i väärtused ning nende abil saab leida $f(x)$.

$f(x)$ leidmiseks on veel puudu üks liige - b. b leidmiseks kasutatakse Karush-Kuhn-Tuckeri tingimusi, mille kohaselt duaalsete muutujate ning optimiseerimisülesande kitsenduste vahelised korrutised peavad võrduma nulliga [12, lk 16] [13, lk 255].

$$\begin{aligned} \alpha_i(\varepsilon + \xi_i - y_i + \langle w, x_i\rangle + b) &= 0 \\ \alpha_i^*(\varepsilon + \xi_i^* + y_i - \langle w, x_i\rangle - b) &= 0 \end{aligned} \tag{10}$$

$$\begin{aligned} (C - \alpha_i)\xi_i &= 0 \\ (C - \alpha_i^*)\xi_i^* &= 0 \end{aligned} \tag{11}$$

Kui leidub selline i, mille korral leidub $\alpha_i \in (0, C)$, siis 11 esimesest võrdusest järeldub, et $\xi_i = 0$ (kuna sulu väärtus tuleb sellisel juhul nulli ja C vahepealne). Seda teadmist ning funktisooni $f(x)$ kuju $f(x) = \langle w, x\rangle + b$ kasutades, saame 10abil, et

$$\varepsilon - y_i + \langle w, x_i\rangle + b = 0 \Longleftrightarrow b = y_i - \langle w, x_i\rangle + \varepsilon.$$

Analoogiline juht kehtib ka juhu $\alpha_i^* \in (0, C)$ jaoks. Sellise lähenemise miinuseks on nõutav indeksi i olemasolu, nii et $\alpha_i^* \in (0, C)$ või $\alpha_i \in (0, C)$ kehtiks.

Võrdustest 10 ning 11 saab teha mitmeid kasulikke järeldusi. Esiteks, ainult objektid (x_i, y_i), mille korral $\xi_i^* > 0$ asuvad väljaspool epsilon-tundetut piirkonda ehk 11 põhjal $\alpha_i^{(*)} = C$. Teiseks, $\alpha_i \alpha_i^* = 0$ ehk α_i ja α_i^* ei saa olla samaaegselt nullist erinevad. Veel üks võimalus vabaliikme b leidmiseks on toodud raamatus [13, ptk 10].

Tingimusest 10 järeldub, et Lagrange'i kordajad α_i ja α_i^* saavad olla nullist erinevad vaid väljaspool epsilon-tundetut piirkonda ehk epsilon-tundetu piirkonna sees saavad α_i, α_i^* väärtuseks nulli. See tuleneb sellest, et $|f(x_i) - y_i| < \varepsilon$ korral on võrrandis 10 teine tegur nullist erinev ning selleks, et Karush-Kuhn-Tuckeri tingimused jääksid täidetuks, peavad α_i, α_i^* olema nullid. Need andmepunktid, mille korral kas α_i või α_i^* on nullist erinevad, on tugivektorid.

2.2.1 Mittelineaarne juht

Kui tahame sobitada sellist regressiooni funktsiooni, mis sõltub andmetest mittelineaarselt, on üheks võimaluseks teisendada andmed mõnda teise (suuremamõõtmelisse) skalaarkorrutisega ruumi $\mathcal{F}$ ning kasutada seal ruumis lineaarset regressiooni. Selleks tutvustatakse tuumafunktsiooni. Tuumafunktsiooni on defineeritud kujul

$$k(x, x') = \langle \varphi(x), \varphi(x)' \rangle \text{ ,} kus\ \varphi : \mathcal{X} \to \mathcal{F}. \tag{12}$$

Tuumafunktsioon annab argumentide skalaarkorrutise pärast nende teisendamist ruuremasse ruumi. Tänu sellele omadusele on optimiseerimine täpselt samasugune nagu varem, kuid lisatud on kujutus φ (skalaarkorrutised on asendatud tuumafunktsiooniga)[14]. Optimiseerimisprobleem 3 saab kuju

$$\begin{aligned} &\text{minimiseerida } \frac{1}{2}\|w^2\| \\ &\text{tingimusel, et } \ |y_i - f(\varphi(x_i))| \leq \varepsilon \end{aligned} \tag{13}$$

Sarnaselt eeltoodule lisatakse ka siin abimuutujad, mis lõdvendavad lubatud eksimuse tingimusi ning probleem 4 saab kuju:

$$\text{minimiseerida } \frac{1}{2}\|w^2\| + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \tag{14}$$

$$\text{tingimusel, et } \begin{cases} y_i - f(\varphi(x_i)) & \leq \varepsilon + \xi_i \\ f(\varphi(x_i)) - y_i - & \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* & \geq 0 \end{cases}$$

Kasutades taas Lagrange'i funktsiooni moodustamist ning tehes läbi eelnevas osas kirjeldatud sammud, jõutakse duaalse probleemini ning optimiseerimise probleemist (10) saab [12, lk 19-20]:

$$\text{maksimiseerida } \begin{cases} -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)k(x_i, x_j) \\ -\varepsilon\sum_{j=1}^{l}(\alpha_i + \alpha_i^*) + \sum_{j=1}^{l} y_i(\alpha_i - \alpha_i^*) \end{cases} . \tag{15}$$

$$\text{tingimusel } \sum_{j=1}^{l}(\alpha_i + \alpha_i^*) = 0 \text{ ja } \alpha_i, \alpha_i^* \in [0, C]$$

Sarnaselt muutuvad ka

$$w = \sum_{j=1}^{l}(\alpha_i + \alpha_i^*)\varphi(x_i)$$
$$f(x) = \sum_{j=1}^{l}(\alpha_i + \alpha_i^*)k(x_i, x) + b.$$

Väikseima tõusuga funktsiooni ei leita enam tunnusvektorite ruumis $\mathcal{X}$, vaid $\mathcal{F}$-s [15].

2.2.2 Levinuimad tuumafunktsioonid

Praktikas enam esinevad tuumafunktsioonid on[12][17]

- polünonomiaalne tuumafunktsioon
  $k(x, y) = (\alpha\langle x, y\rangle + c)^d, d = 1, 2, ...$, kus otsitavad parameetrid on d ehk pülünoomi aste, konstant c ning tõus α

- Gaussi tuumafunktsioon
  $k(x, y) = \exp(-\frac{\| x - y \|)^2}{\sigma^2})$, kus otsitavaks parameetriks on σ

- Sigmoidi tuumafunktsioon
  $k(x, y) = \tanh(\alpha\langle x, y\rangle + c)$, kus on kaks otsitavat parameetrit α ja vabaliikme konstant c

Radiaalne baasifunktsioon tähendab seda, et tuuma väärtus sõltub x-st ja y-st ainult läbi eukleidilise kauguse. Gaussi tuum on tuntuim radiaalse baasiga tuumafunktsioon ning enamasti on kirjanduses radiaalse baasiga tuumafunktsioonile viidates silmas peetud just Gaussi tuumafunktsiooni. Gaussi tuumafunktsiooni parameetri sigma üle-ja alahindamise tõttu tekkivad probleemid on illustreerinud Mušnikov [12, lk 21-23]. Ülehindamise korral on funktsioon tundlik erindite suhtes ja tekib ülesobitamine ning testandmestikul saadav tulemus on oluliselt kehvem treeningandmestiku omast.

2.3 Tarkvara

Käesolev magistritöö kasutab tarkvara R, milles on tugivektormasinatega töötamiseks neli põhilist paketti, millest kaks, *e1071* ja *kernlab*, on tugivektorregressiooni jaoks sobivad. Mõlemal paketil on teise ees selged eelised - *kernlab* võimaldab kasutajal ise tuumafunktsiooni defineerida, kuid *e1071* sisaldab paremaid võimalusi mudeli valikuks. Lisaks on paketid *klaR* ning *svmpath*. Viimased kaks on aeglasemad ning *svmpath* töötab vaid klassifitseerimise ülesande korral [8]. Käesolevas magistritöös kasutatakse tarkvara R paketti *Caret* (**c**lassification **a**nd **re**gression **t**raining), mis koondab endas 25 erinevat paketti (seal hulgas *kernlab* ning *e1071*). Paketid, mida *Caret* sisaldab, paigaldatakse arvutisse jooksvalt, kui kasutusele tuleb mõni vastava paketi käsk. Pakett sisaldab kasulikke funktsioone nii andmete jaotamiseks treening-ning testandmestiku vahel, eeltöötlemiseks kui ka muutujate valiku hõlbustamiseks, meetodeid mudeli parameetrite valikuks ning valimi mitmekordseks võtmiseks ülesobitamise ohu vähendamiseks [11].

2.4 Caret pakett

Käesolevas töös jagatakse andmestik treening- ning testandmestikuks. Mudelite valideerimiseks ning parameetrite valikuks jaotatakse treeningandmed omakorda korduvalt osadeks ning pärast parameetrite leidmist kasutatakse kogu treeningandmestikku parima mudeli sobitamiseks ja seejärel hinnatakse testandmestiku, mida sobitamisel ei kasutatud, põhjal saadud mudeli headust.

Otsitavate parameetrite potentsiaalsete väärtustega luuakse kõikvõimalikud komplektid. Kasutades valimi taasvaliku võtmise meetodeid, proovitakse kõikide kandidaatväärtustega mudeleid igal valimil ettenähtud osal ning hinnatakse mudeli sobivust ülejäänu peal. Taasvaliku (valideerimise) viga hinnatakse, keskmistades veahinnangud üle valimite. Saadud veahinnangute abil leitakse sobivad parameetrite kombinatsioonid. Pärast lõplike väärtuste valimist sobitatakse mudel kogu treeningandmestikule[11].

2.4.1 Paketi Caret funktsioonid ning nende argumendid

'train' funktsiooni saab kasutada mudeli otsitavate parameetrite väärtuste valimiseks, mudeli täpsuse hindamiseks valideerimisel ning taasvaliku meetodi määramiseks. Siinakohal tuuakse tugivektorregressiooni kasutamiseks olulisemad argumendid [11]

x argumenttunnuste maatriks

y funktsioontunnuse väärtused, mille järgi programm saab aru, kas tegemist on klassifitseerimise või regressiooni probleemiga. Kui y on pidev tunnus, sobitatakse regressioonmudeleid, diskreetsel juhul klassifitseerimismudeleid.

method mudeli tüübi täpsustus nt svmLinear, svmPoly, svmRadial

metric määrab, millise näitaja järgi valitakse parim mudel, tugivektorregressooni puhul on selleks vaikimisi ruutkeskmine viga

trControl parameetrite väärtused, valimivõtmise meetod ning iteratsioonide arv; enamasti täpsustatakse selle argumendi väärtused väljaspool *train* funktsiooni eraldi funktsiooni *trainControl* abil

tuneLength määrab parameetrite kombinatsioonide arvu. Vaikimisi on tugivektorregressiooni korral C väärtusteks $0,25; 0,5; 1$. Testitavad väärtused sõltuvad meetodist.
Tunelength'i väärtuse m korral testitakse regressioonimudelite korral C väärtusi $2^{-2}, 2^{-1}...2^{m-3}$. $Tunelength = 5$ korral klassifitseerimise korral on C väärtused 0,1; 1; 10; 100 ning 1000.

2.5 K-kordne ristvalideerimine

Selleks, et hinnata etteantud parameetritega mudeli täpsust uute andmete prognoosimisel, on üheks populaarseks lähenemiseks järgmine protseduur. Andmestik jagatakse enamasti juhuslikult K-ks osaks. Iga osa korral kasutatakse ülejäänud K-1 osa andmeid mudeli sobitamiseks ning leitakse valideerimishinnang vaadeldava osa andmeid kasutades (st sobitamist ja valideerimist sooritatakse kokku K korda). Iga kord on testimiseks võetav osa erinev, mistõttu on meetodi eeliseks see, et kõiki andmeid kasutatakse nii treenimisel kui testimisel (juhul kui vahepeal ümberjagamist ei toimu). Seejärel arvutatakse keskmine viga üle K jagamise. Protseduuri võib korrata R korda ning siis saadakse kokku RxK hinnangut, mis keskmistatakse. Kasutades K-kordset ristvalideerimist, pole valideerimisandmestikku eraldi vaja. Meetodi miinuseks on ajakulukus. Suurte andmestike korral kasutatakse 3-kordsest ristvalideerimisest, tüüpiline valik on 10-kordne ristvalideerimine [5, slaidid 7-9].

2.5.1 Ristvalideerimine aegridade korral

Tarbimise andmed moodustavad aegrea. Aegrea andmed ei vasta juhuslikele katsetele vaid on sageli ajas tugevalt korreleeritud ning kuna prognoosimisel on tähtis just see, kuidas mineviku andmete põhjal tulevikku prognoosida, ei anna enamasti juhuslike valimite abil teostatud ristvalideerimine õiget hinnangut mudeli käitumisele tuleviku prognoosimisel. Seetõttu tuleb kasutada teistsugust taasvaliku meetodit. Aegrea andmete korral saavad treeningandmestiku moodustada vaid testandmestikule eelnevad andmed ja testandmestikuna ei kasutata perioodi algust. Olgu k vähim valimimaht, millega võiks usaldusväärse prognoosi saada, siis üks võimalikke algoritme on järgmine [6]:

1. Valida vaatlusandmed ajahetkel $k + i$ testandmestiku jaoks ning hinnata vaatlustelt 1,2,...,$k + i - 1$ prognoosimudel. Arvutada ajahetkede $k + 1, k + i + 1, \ldots, k + i + m$ prognoosivead

2. Korrata eelmist sammu i = 1,2,...,$T - m$ jaoks, kus T on vaatluste arv.

3. Arvutada varasemalt saadud prognoosivigade pealt prognoosi täpsus.

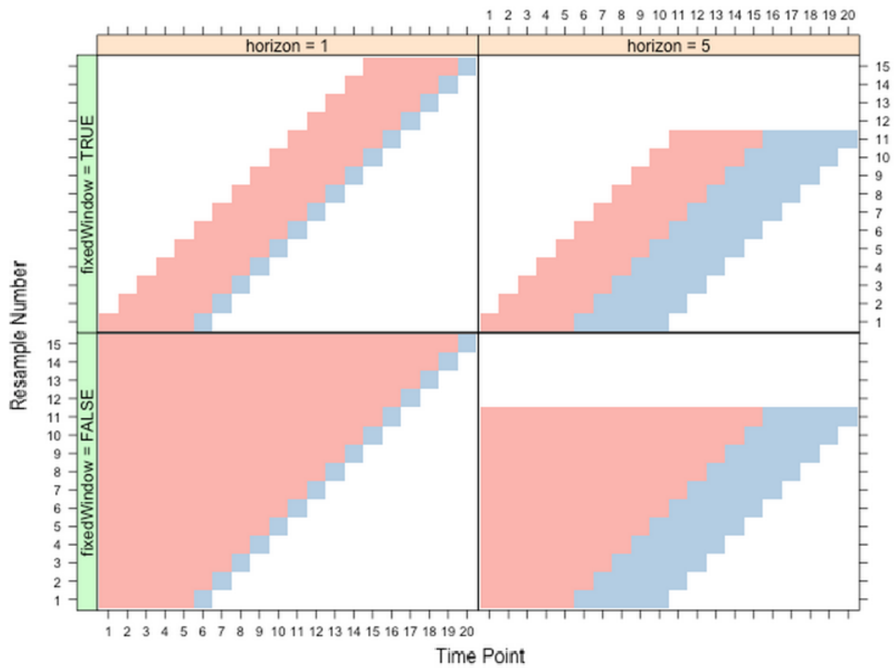
Paketis *Caret* on eespool kirjeldatud algoritmi võimalik rakendada, kasutades funktsioonis *trainControl* argumente [10]:

initialWindow esialgne järjestikuste vaatluste arv treeningvalimis, mida prognoosimiseks kasutatakse

horizon järjestikuste vaatluste arv, mida testvalimis prognoositakse

fixedWindow väärtuse 'FALSE' korral algab treeningvalim alati esimsest vaatlusest ning valimimaht muutub, väärtuse 'TRUE' korral on kasutatavate vaatluste arv fikseeritud.

Joonis 3 illustreerib aegridade valideerimismeetodit. X-teljel on ajahetk ning y-teljel on taasvaliku number. Esialgne vaatluste arv treeningandmestikus on fikseeritud viieks. Vasakus tulbas on toodud juht, kus testandmestikus on 1 vaatlus. Üleval on treeningvalimi maht fikseeritud, all mitte ja kasvab igal sammul. Paremal pool on sarnane situatsioon, kuid testandmestikus prognoositakse 5 vaatlust.



Joonis 3: Näide initialWindow = 5 korral [10]

Teine võimalus aegridadele tugivektorregressiooni sobitamiseks, on ise määrata treeningandmed ja testandmed, vältides juhuslikku valikut ajaliselt seoses olevatel andmetel. Paketis *Caret* saab selleks käsus *trainControl* määrata järgmised argumendid:

index treenimiseks mõeldud andmete reaindeksite list igaks iteratsiooniks

indexOut sama pikkusega reaindeksite list, mis täpsustab testandmestiku igal sammul

2.6 Parameetrite väärtuste analüütiline hindamine

Järgnev ülevaade antakse artikli [2] põhjal. Korduva sobitamise ja valideerimise teel parameetrite väärtuste otsimine on arvutuslikult kallis tegevus. Selle probleemi vältimiseks on välja pakutud mitmeid lahendusi, näiteks alternatiivsete parameetrite leidmine, millele on kergem intuitiivselt väärtust pakkuda või lähendite leidmine analüütiliselt. Üheks variandiks on kasutada parameetrit ν, mille kohta on võimalik täpsemalt lugeda [13, lk 260-266] ning selle parameetri väärtuste valikut käsitletakse artiklis[2]. Käesolevas töös piirdutakse teeoriaosas tutvustatud parameetritega. ε peaks olema seda väiksem, mida suurem on andmestik. C valikul pakutakse üheks variandiks valida C võrdseks prognoositava tunnuse väärtuste ulatusega. Sellise valiku puuduseks on erinditega mittearvestamine. Cherkassky ja Ma [2, lk 5] pakuvad välja parameetri C valikuks seose

$$C = max(\mid \overline{y} + 3\sigma_y \mid, \mid \overline{y} - 3\sigma_y \mid), \tag{16}$$

kus $\overline{y}$ on treeningandmete funktsioontunnuse keskväärtus ning σ_y on sama tunnuse standardhälve. Praktikas tihti treeningandmed tsentreeritakse ehk $\overline{y} = 0$ ja C valiku seos saab kuju $C = 3\sigma_y$. Kui tsentreeritakse, saab C väärtuse 3. Epsiloni valik sõltub valimimahust ning prognoosijääkide suurusest. Suure valimimahu korral soovitakse epsiloni valikuks valemit

$$\varepsilon = \tau\sigma\sqrt{\frac{ln(n)}{n}}, \tag{17}$$

kus n on treeningvalimi maht, konstandi τ väärtuseks soovitatakse vastavalt katsetulemustele 3 ning σ on jääkide standardhälve. Viimast saab treeningvalimi pealt hinnata, sobitades andmetele mingi teise mudeli (näiteks kõrgema järgu polünoomi) ning kasutades valemit:

$$\hat{\sigma}^2 = \frac{1}{n-d}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{18}$$

kus d on parameetrite arv (polünoomi korral aste) ning n on treeningvalimi maht.

2.7 Skaleerimine

Skaleerimine mõjutab tugivektormasinatega saadud tulemust, kuna see mõjutab otseselt erinevatele tunnusvektori väärtustele vastavaid skalaarkorrutisi ning seega ka vaadeldava minimiseerimisülesande 14 lahendit. Seega on andmete eeltöötlemine oluline. Enamkasutatavad moodused on lineaarne teisendus ning normaliseerimine. Tunnuse X lineaarne teisendus lõiku (a,b) käib valemiga

$$z_i = a + \frac{x_i - min(X)}{max(X) - min(X)} * (b - a). \tag{19}$$

Lineaarse teisenduse kitsaskohaks on tundlikkus erindite suhtes. Normaliseerimine käib teisendusega, kus vaatlusest lahutatakse tunnuse keskväärtus ning tulemus jagatkse tunnuse standardhälbega

$$z_i = \frac{x_i - mean(X)}{sd(X)}. \tag{20}$$

Artiklis [16] on uuritud erinevate teisenduste mõju aegridade prognoosimisel ning jõutud järeldusele, et tulemused erinevad aegridade lõikes suurel määral. Eesti elektritarbimise aegrida on sesoonne ning trendita. Eleringi 2011. aasta pikaajalise tarbimise prognoosi (aastani 2025) andmetel on tarbimise kasv aastas keskmiselt 2,2 protsenti[4]. Aastate 2013 ning 2014 andmete põhjal lineaarset trendi arvestada ei saa. Tulemuseks saadakse, et üldiselt eksis lineaarne teisendus lõiku [-0,5...0,5] nii keskmise suhtelise kui ruutvea korral teistest meetoditest arvuliselt vähem ning seega soovitatakse seda, võrreldes teise kahe skaleerimisega. Selgus ka, et Gaussi ning linaarse tuumaga tugivektorregressioon on skaleerimiste lõikes vähemtundlikud kui polünomiaalse tuumaga tugivektorregressioon. Samas sõltusid tulemused suuresti aegrea tüübist ning iga konkreetse aegreaga tuleks valida vastavalt skaleerimisele vähemtundlik tuumafunkstioon. Artikli tulemuste põhjal võiks tarbimise andmetele sobida enim Gaussi tuumafunktsioon ning nii lineaarne skaleerimine kui normeerimine peaksid andma sarnase tulemuse [16]. Artikli tulemusi võetakse käesolevas töös arvesse ning sarnase metoodikaga võrreldakse skaleerimist ja normaliseerimist ka elektritarbimise aegreal.

2.8 Aegrea teooria

Diskreetsete aegridade korral vastavad väärtused võrdsete ajavahemike tagant tehtud mõõtmistele. Ajavahemiku pikkuse h korral mõõdetakse suuruse Z väärtusi ajahetkedel $\tau_i = \tau_0 + ih$, kus $i \in \mathbb{N}$ või $i \in \mathbb{Z}$. Ajamomendile τ_t vastavat juhuslikku suurust Z tähistatakse Z_t.

ARIMA(p,d,q) protsessiks nimetatakse juhuslikku protsessi Z_t, mille d-ndat järku muudud $W_t = (1-B)^d Z_t$, kus B on nihkeoperaator, esituvad kujul

$$\widetilde{W}_t = \sum_{i=1}^{p} \varphi_i \widetilde{W}_{t-i} + A_t - \sum_{i=1}^{q} \theta_i A_{t-i},$$

kus $\widetilde{W}_t = W_t - EW_t$, juhuslikud suurused A_t on sõltumatud sama jaotusega ning tsentreeritud ning on sõltumatud ka suurustest $\widetilde{W}_{t-i}, i = 1, 2, ...$ ARIMA tüüpi mudelite alamklass on perioodiga s multiplikatiivne ARIMA(p,d,q)x(P,D,Q)$_s$ tüüpi mudel. Vastava mudeli kuju on

$$\phi(B)\Phi(B)(1-B)^d(1-B^s)^D Z_t = \theta(B)\Theta(B^s)A_t, \text{kus}$$

$$\phi(x) = 1 - \sum_{i=1}^{p} \phi_i x^i, \Phi(x) = 1 - \sum_{i=1}^{P} \Phi_i x^i$$

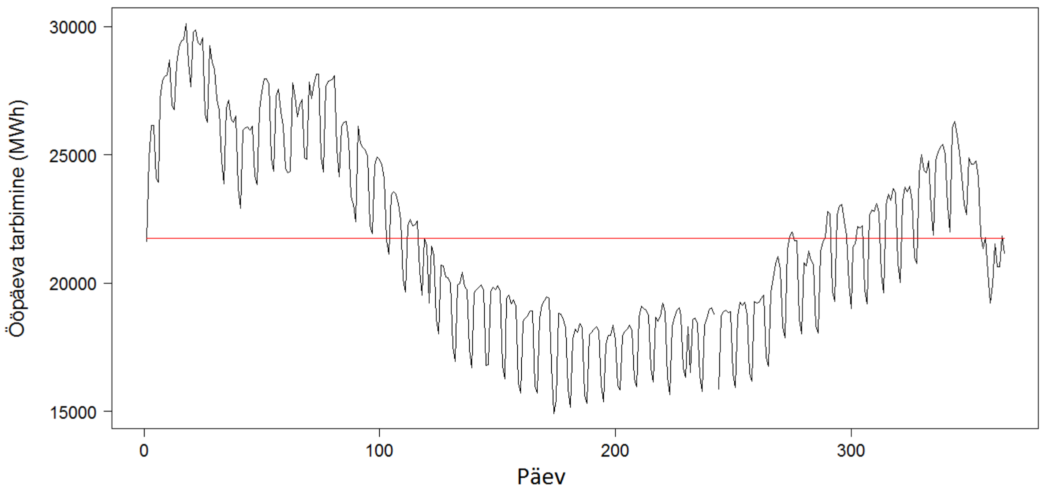
$$\theta(x) = 1 - \sum_{i=1}^{q} \theta_i x^i, \Theta(x) = 1 - \sum_{i=1}^{Q} \Theta_i x^i.$$

Osaline tähendab seda, et ainult osa kordajatest $\varphi_1, \varphi_2, ..., \varphi_p$ ja $\theta_1, \theta_2, ..., \theta_q$ on nullist erinevad [7].

3 Kasutatavad andmestikud

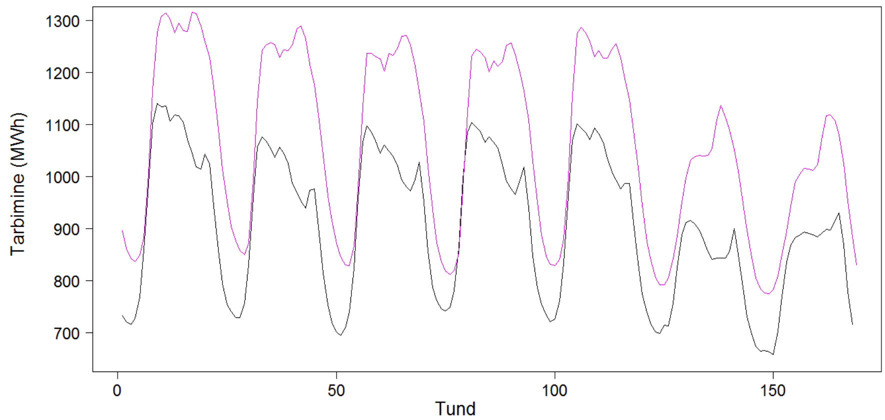
3.1 Andmestike korrastamine ja uurimine

Andmestik sisaldab Eesti elektritarbimise ajalugu tunniajalise täpsusega aastatel 2013 ja 2014 ning 2015 alguses. Esmane samm andmestiku korrastamisel oli imputeerimine. Puuduvad väärtused üksikutel tundidel asendati eelneva ja järgneva tunni tarbimise aritmeetilise keskmisega. Mitme järjestikuse tunni puudumise korral kanti üle 24h varem asetleidnud tarbimine. Ilmaandmestik sisaldas aega, temperatuuri, suhtelist õhuniiskust, õhurõhku, õhurõhu muutust, tuule suunda, tuule kiirust, tuule kiirust puhanguti, pilvisust, nähtusi, sademeid ning nähtavust mõõdetuna 15-20 erinevas ilmajaamas. Ilmaandmestikus käsitleti puudumisi sarnaselt. Kuna üks päev moodustab andmestikust vähem kui 0,2 protsenti ning eelnevad ja järgnevad andmed olid teada, kasutati puuduvate andmete asendamisel keskmistamist. Elektritarbimisest ning selle käitumist ajas on kujutatud joonisel 4. Ööpäevase tarbimise jooniselt on näha, et talveperioodil on tarbimine kõrgem ning suveperioodil madalam. Punane joon illustreerib aasta keskmist tarbimist ööpäeva kohta.



Joonis 4: Tarbimine 2013. aastal

Joonisel 5 on toodud kahe juhuslikult valitud nädala tarbimised. Must joon esindab perioodi 07.04-13.04 2014 ning lilla 12.01-19.01 2015 elektritarbimist tunnise täpsusega.



Joonis 5: Nädala tarbimine

Tarbimise hulk on tööpäevadel kõrgem kui nädalavahetustel ning samuti erineb tarbimist iseloomustava kõvera kuju – tööpäevadel on hommikune tarbimise kasv järsem ning päeva keskel tekib veel üks tipp, võrreldes nädalavahetusega. Tarbimise kõverad erinevad ka sõltuvalt aastaajast, mis on põhjustatud valge aja ning temperatuuri muutumisest.

3.2 Mudeli koostamine

Treeningandmestikus on 2013. ja 2014. aasta elektritarbimise andmed tunniajalise täpsusega. Testandmestiku moodustavad 2015. aasta jaanuar ning veebruar. Eesmärgiks on eelkõige välja selgitada, kas tugivektorregressioon annab eelise teiste meetodite ees, püüdes jõuda võimalikult hea mudelini.

3.2.1 Tunnuste valik

Tuginedes eespool vaadeldud joonistele, kujunesid välja tarbimist potentsiaalselt mõjutavad tunnused:

- **Ilm**- Eesti kliimas mängib temperatur olulist rolli, kuna jahedamal ajal lisandub küttega kaasnev tarbimine. Temperatuuri puhul on oluline arvestada, et temperatuur ei mõjuta tarbimist lineaarselt. Mõlemas suunas ekstremaalsete tingimuste korral on tarbimine suurem. Temperatuuri käitumist saab lähendada tükiti lineaarse funktsiooniga, mis on eriti oluline aegridade mudelite juures. Lisaks võiksid mõjutada veel mõned teised ilmanäitajad, näiteks tuule tugevus.

- **Päev**- nädala sees on elektritarbimine suurem kui nädalavahetusel, mis on otseselt seotud tööl käimisega ning ettevõtete poolt tarbitava elektrikogusega
- **Tund**- tund ehk kellaaeg. Öösel on elektritarbimine väiksem kui päeval. Tarbimise muutumine on seotud ka tavalise tööpäeva pikkusega- tarbimine kasvab kõige järsemalt kella 9 kandis ning langeb 18 juures.
- **Riigipühad**- riigipühad sarnanevad tarbimiselt rohkem nädalavahetustele kui tööpäevadele

3.3 Prognoosi headuse mõõdikud

Erinevate mudelitega saadud prognooside täpsuse võrdlemiseks on võimalik kasutada mitmeid mõõdikuid. Käesolevas magistritöös kasutatakse järgmiseid:

- Keskmine suhteline viga (***m**ean **a**bsolute **p**ercentage **e**rror*)

$$MAPE = \frac{1}{l}\sum_{i=1}^{l}\frac{| y_i - f(x_i) |}{y_i}$$

- Keskmine ruutviga (***r**oot* **m**ean **s**quare **e**rror)

$$RMSE = \sqrt{\frac{1}{l}\sum_{i=1}^{l}(y_i - f(x_i))^2}$$

Keskmine suhteline viga valiti peamiseks headusemõõdikuks, kuna käesoleva töö teemalises kirjanduses oli see kõige laialdasemalt levinud. Treeningandmete põhjal sobitatakse mudelid ning testandmete põhjal, mida sobitamise protseduuris ei kasutata, arvutatakse headusemõõdikud. Täiendava headusemõõdikuna on keskmise suhtelise vea kõrval toodud keskmine ruutviga. Keskmise ruutvea eeliseks on suuremate eksimuste korral veahinnangu ulatuslik kasv, mis on oluline, kuna elektritarbimise andmete korral ei ole kulude suhe võrdne eksimuste suhtega. Näiteks 100 MWh prognoosivea korral on kulutused enam kui kaks korda suuremad, kui 50 MWh eksimuse korral. Lisaks hindab kasutatav tarkvara mudeli sobitamise protsessis parameetrite kombinatsioonide headust keskmise ruutvea abil.

4 Lineaarse regressiooni ning aegrea mudelid

4.1 Mudel 1: ühe tunni prognoos

Esmalt konstrueeritakse lineaarne mudel ühe tunni tarbimise prognoosimiseks. Kuna üks tund on väga lühike aeg, mida ette ennustada, kasutatakse tarbimise ajaloona kõige viimaseid teadaolevaid andmeid ehk eelmise tunni tarbimist ning teise argumenttunnusena tundi. Mudeli üldkuju on järgmine:

$$\text{tarbimine}_i = a * \text{eelmine}_i + c_{i0} + \sum_{j=1}^{23} c_j T_{ij} + \varepsilon_i,$$
$$\text{kus } T_{ij} = \begin{cases} 1, & tund_i = j \\ 0, & \text{muidu} \end{cases}$$

Mudeli keskmine suhteline viga perioodil jaanuar-veebruar 2015 on 0,0167. Temperatuuri lisamine lineaarse regressorina antud mudelit ei paranda, kuna teperatuuri mõju sisaldub juba eelmise tunni tarbimises- keskmine suhteline viga muutub 0,0173-ks. Keskmine ruutviga on 24,07, temperatuuri lisamisel 25,43. Järelduste tegemiseks jäädakse esimese variandi juurde. Selgus, et andmestikus esines kaks viga ning analüütiku soovitusel lisatati 7. veebruari 2. ja 6. tunni tarbimisele 68 MWh. Edaspidises analüüsis kasutatakse korrigeeritud andmestikku.

4.2 Mudel 2: suhetega mudel

Esmaste lineaarsete mudelite eesmärk on välja uurida, milliseid tunnuseid võiks elektritarbimise prognoosiks kasutada ning nende abil fikseerida lävend prognoosiveale, millega tugivektorregressiooni sooritust võrrelda.

Mudel 2 kasutab argumenttunnustena eelmise ööpäeva summarset tarbimist ning kahte indikaatorit ja tundide suhteid. Mudeli kuju on järgmine:

$$\text{tarbimine}_i = (c_1 \text{eelneva 24h tarbimine}_i + c_2 I_E + c_3 I_L + c_4) * C_h + \varepsilon_i$$

Indikaator I_E on defineeritud

$$I_E = \begin{cases} 1, & \text{kui nädalapäev on esmaspäev} \\ 0, & \text{muidu} \end{cases}$$

Analoogiliselt on defineeritud I_L laupäeva kohta. Need indikaatorid tähistavad tööpäevade ja nädalavahetuse üleminekut, kuid kuna üleminekud pole omavahel ühesugused (s.t. tarbimine ei lange laupäeval, võrreldes reedega sama palju, kui tõuseb esmaspäeval, võrreldes pühapäevaga), on indikaatoreid kaks. Kui muutuste iseloomud oleks sarnased, saaks teha ühe tunnuse väärtustega -1, 0 ja 1. Tundide suhted C_h arvutati nädal aega varem esinenud tarbimise pealt (vastava tunni tarbimine jagatud kogu ööpäeva tarbimisega).

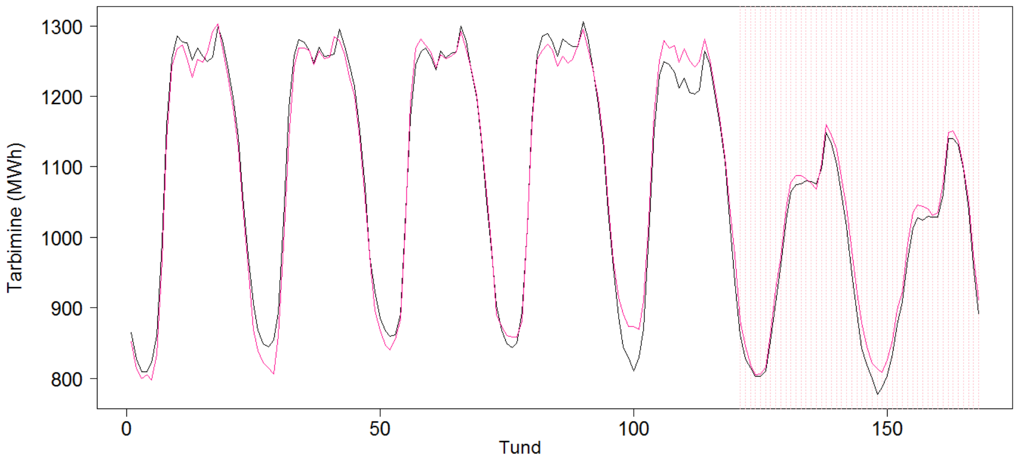
Treeningandmete põhjal saame parameetri c_1 väärtuseks 0,9970, mis tähendab seda, et prognoosimisel kasutatakse 99,7 protsendi ulatuses eelmise ööpäeva tarbimist korrutatuna nädalataguse suhtega. Vähem kui poole protsendi ulatuses tehakse korrektuure nädalavahetuste üleminekute juures. Kordajad c_2 ning c_3 saavad väärtusteks vastavalt 2790,1 ja -2265,2 ehk keskmiselt (kui suhe on 1/24) liidetakse esmaspäeval ühe tunni tarbimisele 116 MWh ning laupäeval lahutatakse tunnise tarbimise koguselt 94 MWh.

Mudeli keskmine suhteline viga jaanuari ning veebruari lõikes on 0,0337. MAPE arvutamisel jaanuari esimest nädalat välja jättes kahaneb veahinnang 0,0286ni. Lisaks 24. veebruari prognoosivigu välja jättes tuleb keskmine suhteline viga 0,0281. Keskmine ruutviga tuli üle kahe kuu 50,23.

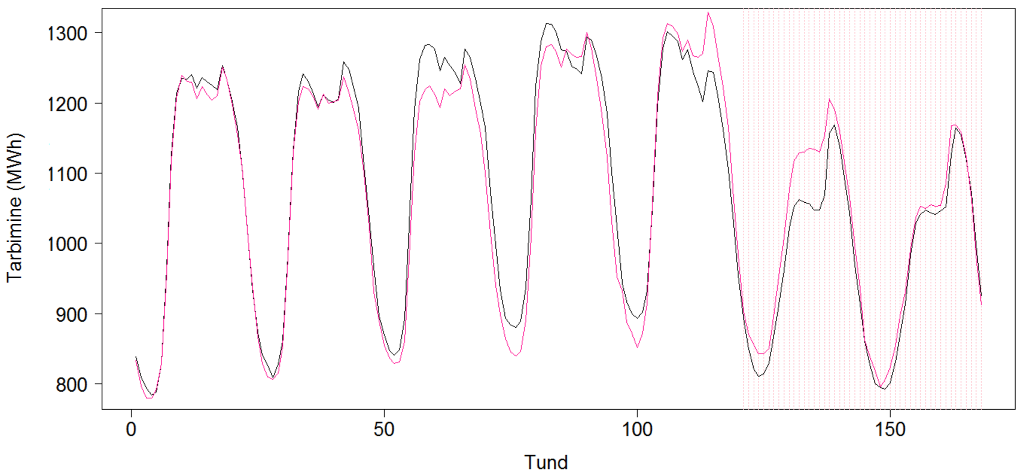
Antud prognoosi korral vaadeldakse mudeli esinemist ka juhul, kui välja jätta ka 8-9. jaanuar, kuna käesoleva aasta alguses olid n.ö. pikad pühad ehk 2. jaanuar sattus reedesele päevale (paljud inimesed tööl ei käinud) ning kuna see prognoos kasutab nädalataguseid suhteid, siis hinnatakse 8. ja 9. jaanuarile puhkepäeva kuju. Tulemuseks saadakse, et mudeli keskmine suhteline viga on esimese nädalata 0,0253. Suuremate veaallikate väljajätmine vea hindamisel ei anna täielikku ülevaadet mudeli headusest, kuid nende veaallikate mõju saab vähendada, mille juurde pöördutakse tagasi "Mudelite parandamise"peatükis.

Joonisel 6 on tegelik tarbimine (must) koos prognoosiga (roosa) perioodil 26. jaanuar kuni 1. veebruar. Heledama roosa taustal on nädalavahetuse päevad. Jooniselt on näha, et toodud nädalal prognoosib mudel küllaltki täpselt. Kõige suurem eksimus tuleb reedesel päeval. Võrdluseks joonistatakse ka järgneva nädala graafik (2-8 veebruar).

Joonis 7 peegeldab, kuidas mudel võtab seoses nädalatagustelt andmetelt arvutatud suhetega üle nädalataguse kuju. Laupäeval, 7. veebruaril on tegemist 31. jaanuari kujuga, mis ei ole enam päeva esimesel poolel sobilik.



Joonis 6: Nädala tarbimise prognoos 26.01-01.02



Joonis 7: Nädala tarbimise prognoos 02.02-08.02

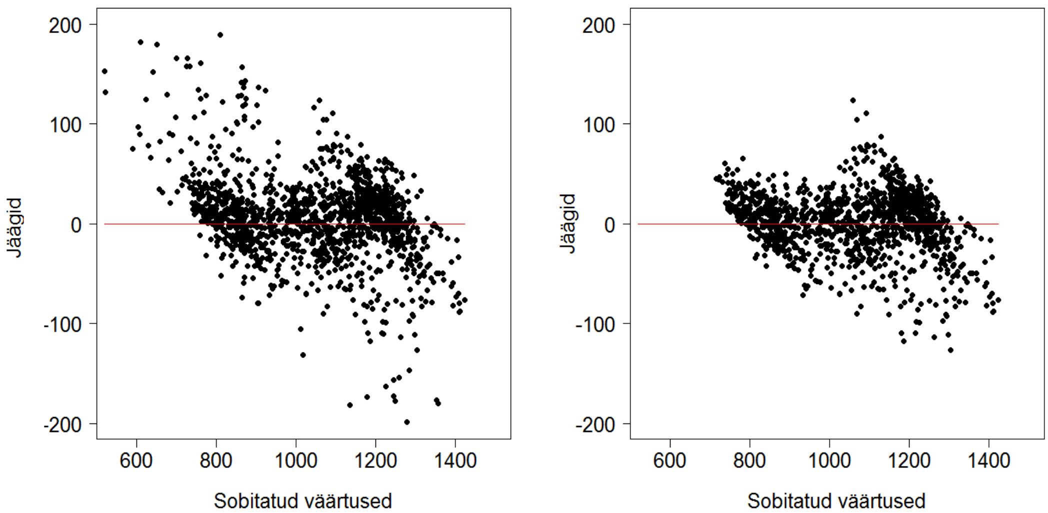
4.3 Mudel 3: koosmõjuga mudel

Eelmise mudeli korral eeldati, et eelmise ööpäeva tarbimise jaotus tundide vahel ei anna infot järgmise päeva tarbimise jaotuse kohta. Samuti eeldati, et kogutarbimise muutumisel jäävad tarbimise jaotumisel tundide vahel proportsioonid samaks. Alternatiiviks on eeldada, et järgmise päeva iga tunni tarbimist saab prognoosida eelmise päeva sama tunni tarbimise kaudu, korrigeerides seda ennustatavast nädalapäevast ja tunnist sõltuva liidetavaga. Nädalapäeva ning tunni koosmõju vaadeldakse, kuna tarbimiskõvera kuju on nädala sees ning nädalavahetusel erinev. 24 tundi tagasi aset leidnud tarbimise hulga arvestamine sisaldab endas juba kaudselt informatsiooni ilmastikuolude kohta. Vastavaks matemaatiliseks mudeliks on

$$\text{tarbimine}_i = c_1 \text{ tarbimine 24 h tagasi}_i + c_2\text{päev}_i * \text{tund}_i + \varepsilon_i.$$

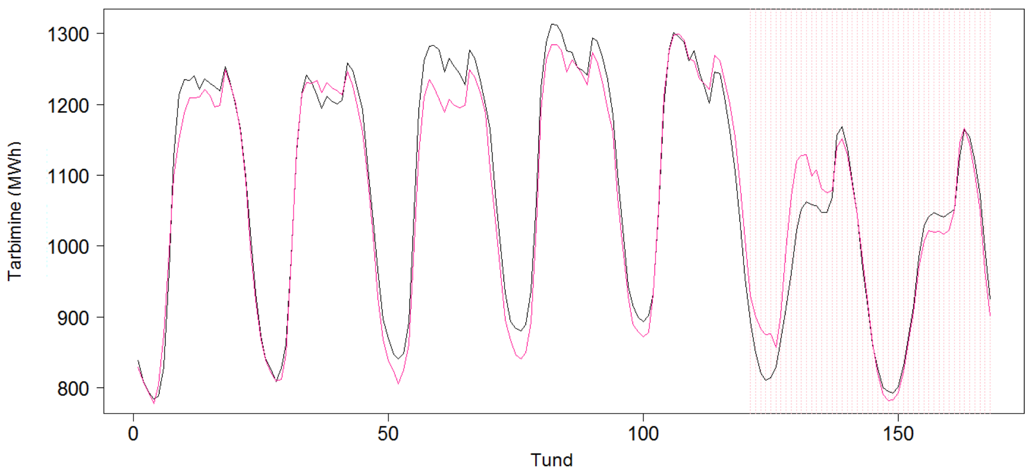
Antud mudelis on päev ja tund faktortunnused. Iga testperioodi päeva jaoks prognoositakse järgmise ööpäeva tarbimine, kasutades prognoosi momendiks olemasolevaid andmeid. Seega tarbimine 24 tundi tagasi on tegelik tarbimine ning prognoosiviga ei ole kuhjuv.

Vaadates jääkide ning sobitatud väärtuste vahelist graafikut (joonis 8) selgub, et prognoosides on testperioodil süstemaatilisi vigu. Nimelt suurte väärtuste prognoosimisel kipub mudel tegelikke väärtuseid pigem ülehindama ja väikeste väärtuste prognoosimisel pigem alahindama. Samas on selge, et mudel ei sisalda infot nädala sisse jäävate riigipühade kohta ning kuna riigipühal on tarbimise struktuur ja kogus oluliselt tööpäeva omast erinevad, ülehinnatakse tööpäevale sattuva riigipüha puhul tarbimist vähemalt tööaja sisse jäävate tundide puhul ning järgneval päeval alahinnatakse tarbimist tööajal. Parempoolsel joonisel on eemaldatud jaanuari esimene nädal ning 24. veebruar koos järgneva päevaga. Tendents suuremaid tarbimisi ülehinnata jääb alles, kuid ei paista enam nii suurel määral silma.



Joonis 8: Jääkide graafik

Lihtsama jälgitavuse huvides kuvatakse joonisel 9 ühte nädalat prognoosimise tulemustest. Joonisel on tegelik tarbimine (must) koos prognoosiga (roosa) perioodil 02.02 kuni 08.02. Heledama roosa taustal on nädalavahetuse päevad.



Joonis 9: Nädala 02.02-08.02 prognoos

Mudel prognoosib kõige paremini teisipäevad ning kolmapäevad, sest esmaspäev, teisipäev ning kolmapäev on omavahel kõige sarnasemad päevad. Suurimad probleemid 24-tunni-taguse tarbimise kasutamisel on üleminekud tööpäevade ning nädalapäevade vahel ning see, et erindid põhjustavad suure vea järgmisel päeval. Laupäeval on mudeli prognoosi järgi paar sakki lisaks, mida koosmõju kordajad ära ei silu ning esmaspäeva kujust puuduvad nädalavahetusele omaselt hommikused võnkumised.

Koosmõjuga lineaarse mudeli keskmine suhteline viga on jaanuaris-veebruaris 0,0301. MAPE arvutamisel jaanuari esimest nädalat välja jättes kahaneb veahinnang oluliselt 0,0243-ni. Lisaks 24. veebruari prognoosivigu välja jättes tuleb keskmine suhteline viga 0,0236. Keskmine ruutviga üle kogu perioodi on 46,90.

Keskmine eksimus on 5,743 MWh. Kõige rohkem eksib mudel ajavahemikus 6-12, mil tarbimine on kasvav. Suurim eksimus on 269,14 MWh, mis esineb 1. jaanuaril kell 9 ning on seletatav sellega, et 31. jaanuar oli ametlikult tööpäev ning selle päeva kella 9 tarbimine võeti 1. jaanuari kella 9 prognoosimisel aluseks. Kõige enam eksib mudel absoluutväärtuselt keskmiselt 15nda tunni prognoosimisel ning kõige vähem 2. tunni korral. Protsentuaalselt on suurim eksimus 9ndal tunnil ning väikseim 23. tunnil. Seega võib öelda, et mudel eksib pigem suuremate tarbimiste korral.

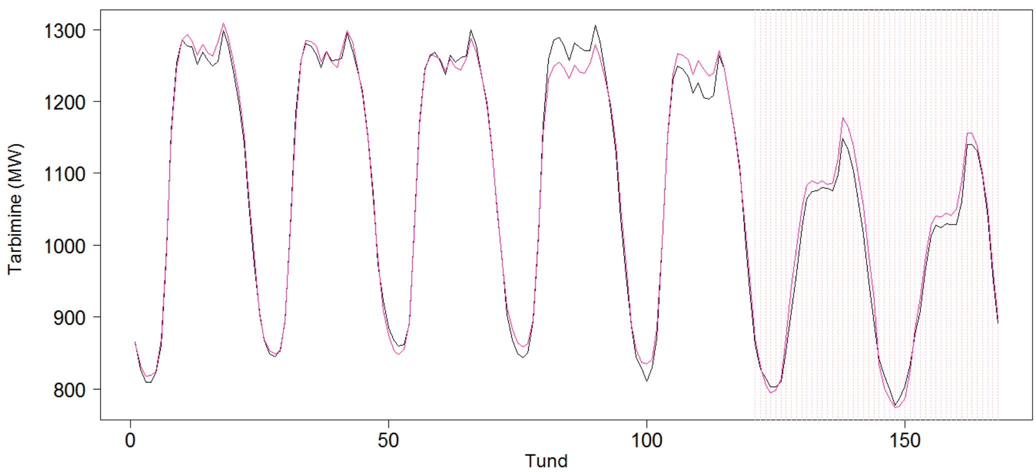
4.4 Mudel 4. Aegrida

On selge, et elektritarbimise andmete vahel on tugev ajaline sõltuvus, mistõttu tasub võrdluse mõttes uurida ka mudeleid, mis seda arvestavad. Üheks populaarseks selliste mudelite klassiks on aegridade analüüsist pärit ARIMA mudelid. Kuna vaadeldava aegrea puhul ei ole standardsed mudeli otsimise protseduurid mitme erineva perioodilisuse (ööpäevane, nädalane, aastane) tõttu hästi kasutatavad, piirdume võimalikult lihtsa mudeliga, kasutades vaid aegrea andmeid, kuid mis siiski arvestab erinevaid perioode.

Andmetele sobitatakse osaline multiplikatiivne ARIMA(25,0,25)x$(1,1,1)_{168}$. Saadud mudeliks on

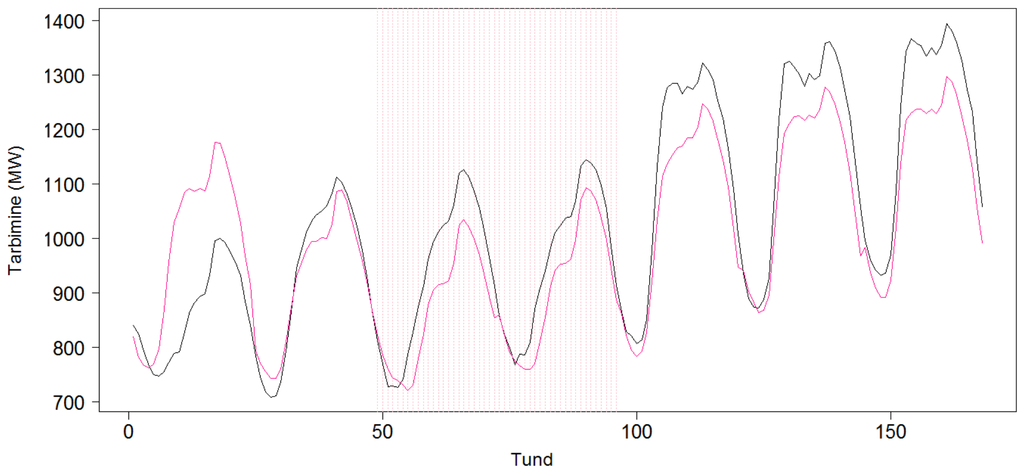
$$\begin{aligned}Z_t = {} & 0,98Z_{t-1} + 0,94Z_{t-23} + 0,94Z_{t-24} - 0,87Z_{t-25} + 2,09Z_{t-168} - \\ & 1,95Z_{t-169} - 1,02Z_{t-191} - 1,98Z_{t-192} + 1,82Z_{t-193} - 0,09Z_{t-336} + \\ & 0,09Z_{t-337} + 0,0008Z_{t-359} + 0,08Z_{t-360} - 0,08Z_{t-361} - A_t - \\ & 0,021A_{t-1} + 0,05A_{t-23} - 0,69A_{t-24} + 0,06A_{t-25} + -0,86A_{t-168} + \\ & 0,02A_{t-169} - 0,05A_{t-191} + 0,59A_{t-192} - 0,05A_{t-193}\end{aligned}$$

Üle testperioodi jaanuar-veebruar on ARIMA mudeli keskmiseks suhteliseks veaks 0,0284 ning jaanuari esimese nädalata 0,0227. Keskmine ruutviga on 45,99.



Joonis 10: Nädala 26.01-02.02 prognoos osalise sesoonse multiplikatiivse ARIMA mudeliga

Joonisel 10 on toodud 26.01-02.02 prognoos koos tegeliku tarbimisega. Jooniselt on näha, kuidas prognoosimine on üldiselt hea, kuid joonis 11 peegeldab, kuidas riigpühasid mudel hästi ei ennusta ning esimene pühade päev on kõige halvemini prognoositud.



Joonis 11: Nädala 01.01-07.01 prognoos osalise sesoonse multiplikatiivse ARIMA mudeliga

4.5 Esimene tugivektorregressiooni mudel

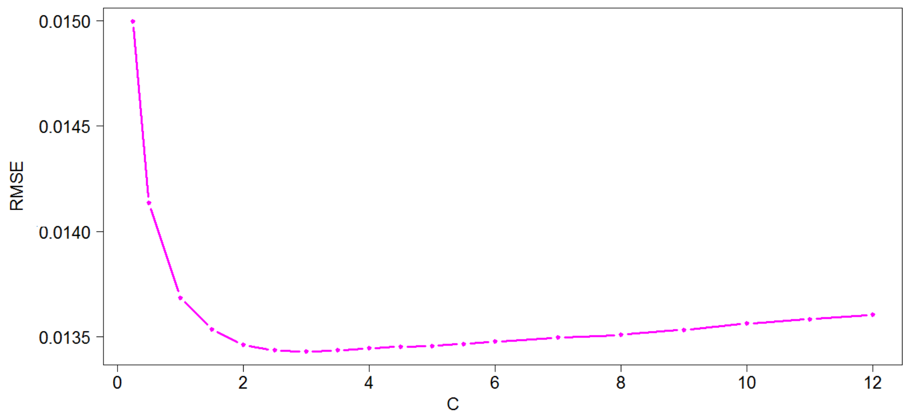
Kirjandust uurides selgub, et tugivektorregressiooni abil elektritarbimise prognoosimine on muutumas üha populaarsemaks. Artikkel [1] keskendub lühiajaliseks prognoosimiseks võimalikult lihtsa mudeli leidmisele. Kasutades argumenttunnustena viimase 24 tunni tarbimisi, nädalapäeva ning tundi, leitakse mudel, mis ennustab järgmise 24h tarbimise. Esmalt ennustatakse esimese tunni tarbimine ning selle abil rekursiivselt järgmise 23 tunni tarbimised. Analüüsitud mudeli treeningandmeteks võeti detsember, jaanuar ning veebruar aastast 2000, mis pärinesid Uus-Inglismaa regioonist. Selliselt tegutsedes saavutati keskmine suhteline viga 2,16 protsenti ühe nädala kohta jaanuaris 2001. Eelpool mainitud artikli põhjal proovis Serres [14] prognoosida New Yorki elektritarbimist. Varasemalt oli New Yorki osariik jagatud 11 piirkonnaks ning igas neist sooritati sõltumatu prognoos. Seni kasutati kunstlike närvivõrkude meetodit, mis võimaldas keerukat andmete struktuuri. Andmeid treeniti varasema 3-4 aasta põhjal ning argumenttunnuseid oli palju: ajaloolised tarbimise andmed, 7 ilmastikunäitajat, pühade indikaator ning kuu aja lõikes konstantsed olevad majandusnäitajad. Kogu tarbimise prognoosiks summeeriti kõikides piirkondades saadud tulemused. Tugivektorregressiooni puhul kasutas Serres treenimiseks 8760 juhuslikku tundi aastatest 2010-2013 ning testmiseks 2014. aasta alguse andmeid. Ühe nädala kohta saavutati keskmine suhteline viga 3,13 protsenti, kuid pikemaajaliselt (kuni juunini) ennustades tõusis suhteline viga 5,24-ni, mis oli suurem kui kunstlike närvivõrkudega saavutatud suhteline viga. Saadud tulemus ei olnud piisavalt hea, kuid autori arvates võiks mõningate ilmanäitajate lisamisel ning samuti piirkondades erinevate mudelite tegemisel saada varasema meetodiga vähemalt sama hea tulemuse. Mudel on lihtsam, saadud kergema vaevaga ja vähem keeruka andmestruktuuriga.

Esimesena prognoositakse tugivektorregressiooniga Eesti elektritarbimine analoogselt artiklites [1] ja [14] kirjeldatud meetodiga. Andmestikku luuakse tunnused 'tagasi1' kuni 'tagasi24', mis on vastavalt tarbimine 1 tund tagasi, 2 tundi tagasi kuni 24 tundi tagasi. Lisaks on argumenttunnusteks nädalapäev ning tund. Sarnaselt artiklitest loetuga prognoositakse ühe tunni tarbimine ning järgnevad 23 tundi leitakse sama mudeli abil rekursiivselt. Enne regressioonimudeli hinamist skaleeritakse andmed lõiku [0,1]. Tuumafunktsiooniks valitakse Gaussi tuum. Ristvalideerimise korral kasutatakse 10-kordset ristvalideerimist, mis osutub ajakulukaks ning aja kokkuhoiu mõttes minnakse parameetrite erinevate väärtuste katsetamisel üle 5-kordsele ristvalideerimisele kahe kordusega.

Sobitades mudeli käsuga *train*, otsitakse parimat mudelit parameetritega $\varepsilon =$ 0,1, $C = (0,25; 0,5; 1)$, $\sigma =$ 0,0413 (hinnatud andmetelt) hulgast, kasutades ristvalideerimist. Parimaks C väärtuseks osutub $C = 1$ ning vastava mudeli viga testandmetel on 0,00609. Mudel kasutab 2241 tugivektorit. Leitud mudeli keskmine suhteline viga testandmetel jaanuar-veebruar on 0,0308, pärast esimese nädala väljaarvamist 0,0268.

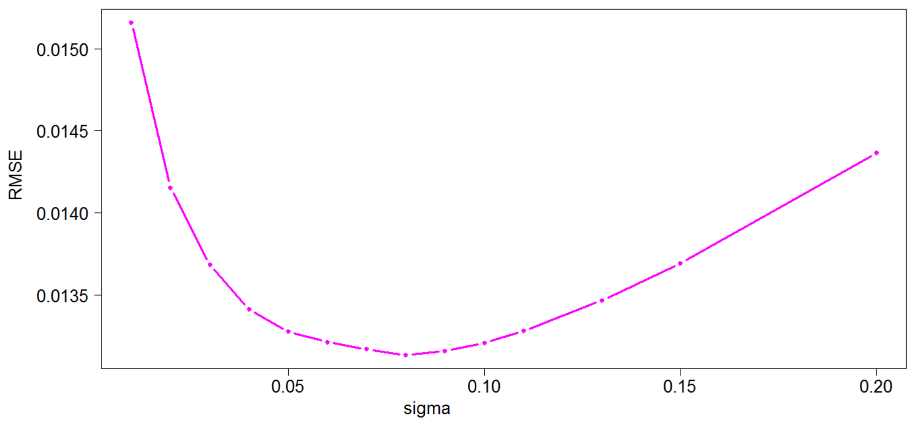
Eelneval sobitamisel otsiti parimat mudelit väga väikese arvu erinevate parameetritega mudelite hulgast (võrreldi ainult kolme). Seetõttu pakub huvi, kas tulemust on võimalik oluliselt prandada, kui laiendada vaadeldavate parameetrite väärtuste hulka. Esialgu jäetakse ε paika 0,1-ks ning fikseeritakse andmetelt hinnatud $\sigma = 0,0413$. Regulatsiooniparameetrile C antakse väärtuseid vahemikus (0,5...6) sammuga 0,5 ning vahemikus (6,...,12) sammuga 1.

Jooniselt 12 on näha, et C kasvades kuni kolmeni langeb keskmine ruutviga küllaltki kiiresti. Edaspidi on C suurenedes ruutkeskmise vea kasvamismäär aeglasem kui kahanemismäär väiksemate väärtuste korral. Väikseim keskmine ruutviga saadakse treeningandmestikul parameetri C väärtuse 3 korral.



Joonis 12: Keskmise ruutvea käitumine parameetri C erinevate väärtuste korral

Sarnaselt anti sigmale erinevaid väärtusi. Joonisel 13 on kujutatud keskmise ruutvea käitumine erinevate sigma väärtuste korral. Keskmise ruutvea mõttes on parim sigma väärtus 0,08.



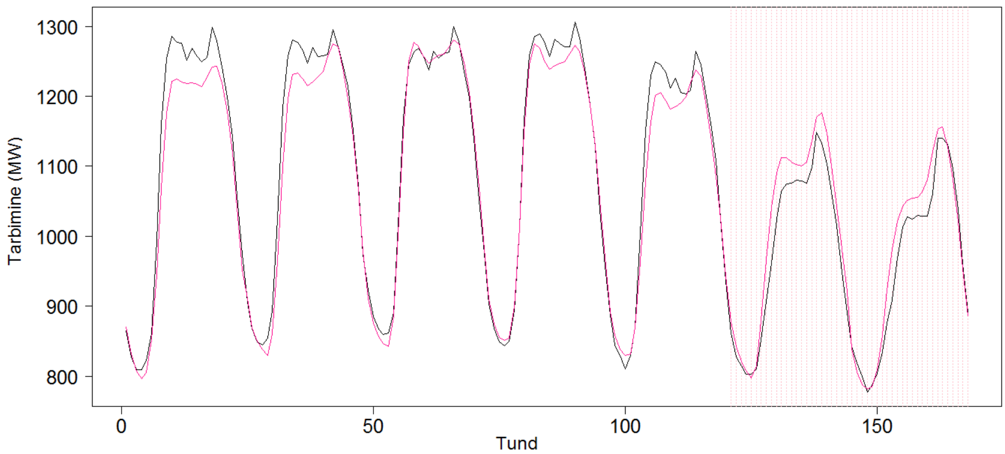
Joonis 13: Keskmise ruutvea käitumine muutuva sigma korral

Parameetrite väärtustega $C = 3$ ning $\sigma = 0,08$ tehakse mudel kogu treeningandmestikule. Tulemust parandatakse veel parameetri epsilon väärtuse vähendamisega, kuna väärtuse $\varepsilon = 0,1$ korral ei peeta oluliseks eksimusi kuni ligikaudu 120 MWh, mis on tunnise tarbimise korral liiga suur. Epsilon fikseeritakse 0,02ks. Parameetri sigma väärtuseks hinnatakse andmetelt $\sigma = 0,0413$. Parameetrile C omistatakse võimalikud väärtused 0,25 ning 0,5 ja $1, 2, \ldots, 6$. Tulemused on toodud tabelis 1. Põhjalikuma parameetrite otsimisega õnnestus nii keskmist suhtelist viga kui ruutviga oluliselt vähendada.

	C	σ	ε	MAPE Jaan-veebr	MAPE nädalata	RMSE	Tugi-vektoreid
Vaikeparameetrid	1	0,0413	0,1	0,0308	0,0268	44,58	2243
C, σ muutmisel	3	0,08	0,1	0,0285	0,0252	41,13	1823
ε muutmisel	6	0,0413	0,02	0,022	0,019	34,27	10607

Tabel 1: Esimest tugivektorregressiooni mudelit iseloomustavad suurused erinevate parameetrite väärtuste komplektide korral

Joonisel 14 on toodud tarbimine (must) ning selle prognoos(roosa) ajavahemikus 26.01-01.02. Prognoos on leitud parima mudeli, parameetri $\varepsilon = 0,02$ korral, abil. Heledama roosa taustal on nädalavahetus.



Joonis 14: Nädala prognoos 20.01-01.02

Parimaks parameetrite kombinatsiooniks valitakse $(C; \sigma; \varepsilon) = (6; 0,0413; 0,02)$. Artiklis [1] saavutatud keskmine suhteline viga ühe nädala kohta on 2,16 protsenti. Eesti väikseim keskmine suhteline viga protsentides ühe nädala kohta on 1,49, suurim 4,73 (jaanuari esimene nädal) ning on keskmiselt ligikaudu 2,3. Serres [14] arvutas analoogse mudeliga keskmise suhtelise vea ühe nädala kohta 3,13 protsenti ning pikema perioodi peale 5,24. Lisaks proovib Serres sama meetodiga 40 tunni prognoosimist ning saab keskmiseks suhteliseks veaks nädala kohta 5,48 protsenti. Eesti elektritarbimise kõvera kuju on nädalavahetusel ning tööpäevadel erinevam kui artiklites [1] ja [14] ning tarbimise kogus on üle 10 korra väiksem, kuid võib öelda, et mudeli sooritus on sarnane või isegi parem.

Enne viimase mudeli rekursiivse osa rakendamist prognoositi 1 tunni tarbimist võrdlemaks esimese linaarse mudeliga. Tugivektorregressiooni viimase mudeli keskmine suhteline viga ühetunnise prognoosi korral on 0,0080 lineaarse mudeli 0,0163 vastu. Seega võib öelda, et ühe tunni prognoosimisel on tugivektorregresiooni mudel täpsem. Samas tuleb silmas pidada, et esimeses lineaarses mudelis on vaid kaks argumenttunnust.

4.6 Teine tugivektorregressiooni mudel

4.6.1 Analüütiliste lähenditega

Antud tugivektorregressiooni mudel on lineaarse mudeli number 2 analoog, mis tähendab, et argumenttunnusteks on eelneva ööpäeva tarbimine ning kaks indikaatorit, mis iseloomustavad tööpäeva ning nädalavahetuse üleminekut, kõik korrutatuna nädal varem arvutatud tundide suhetega:

$$tarbimine = (c_1 eelneva\, 24h\, tarbimine + c_2 I_E + c_3 I_L + c_4) * C_h + \varepsilon$$

Andmestik jagatakse järgmiselt: treeningandmete hulga moodustastavad 2013. aasta ning jaanuar-september 2014, valideerimisandmesikku kuuluvad oktoober-detsember 2014 ning testandmestikku jaanuar-veebruar 2015. Andmed normaliseeritakse valemi 20 järgi ning arvutatakse parameetrite C ja ε väärtused. Normaliseerimisest tulenevalt on funktsioontunnuse keskväärtus $\overline{y} = 0$ ning standardhälve $\sigma_y = 1$ ja valemi 16 järgi saab parameeter C väärtuseks 3. Tuumafunktsiooniks valitakse Gaussi tuum. Valemi 18 abil prognoosivea standardhälbe arvutamiseks sobitame treeningandmetele kümnenda astme polünoomi abil mudeli

$$\text{tarbimine} = \beta_0 + \beta_1 * (\text{korrutis2}) + \beta_2 * (\text{korrutis2})^2 + ... + \beta_{10} * (\text{korrutis2})^{10} + \varepsilon.$$

Saadud mudeli prognoosivigade standardhälbeks on $\sigma = 0{,}3073$. Valemi 17 kohaselt leitakse ε väärtus, milleks on 0,0231. Fikseeritud $\varepsilon = 0,0231$ ning $C = 3$ korral otsitakse nii keskmise suhtelise vea kui keskmise ruutvea põhjal optimaalseimat σ väärtust. Pärast optimaalseima sigma väärtuse leidmist fikseeritakse see ning antakse parameetrite C ja ε väärtusteks kõikvõimalikud paarid $(C; \varepsilon)$ valemite abil arvutatud lähendite mõistlikus ümbrusest. Korraga testitakse 40-50 erinevat parameetrite kombinatsiooni. Kui valituks osutuvad määratud vahemike otspunktides asuvad väärtused, laiendatakse ümbrusi.Vajadusel vähendatakse ühe parameetri muutumise sammu. Optimaalseimat $(C; \varepsilon)$ paari fikseeritud σ korral otsitakse samuti nii keskmise suhtelise vea kui ruutvea järgi.

Võrdluseks teisendatakse andmed vastavalt valemile 19 lineaarselt lõiku [-0,5...0,5]. Andmestik jagatakse treening-, valideerimis- ning testandmestikuks sarnaselt normaliseerimisega ning leitakse valemi 16 abil C ning varemleitud prgnoosijääkide standardhälvet kasutades valemi 17 abil ε väärtused ning hinnatakse sigma. Pärast fikseeritakse saadud sigma väärtus ning leitakse parameetrite väärtuste paaride abil uuesti C ja ε. Kõiki otsinguid tehakse nii keskmise suhtelise vea kui keskmise ruutvea suhtes. Saadud tulemused on kujutatud tabelis 2.

	Headuse mõõdik		Väärtused			MAPE	RMSE
	σ	C ja ε	σ	C	ε	Jaan-veebr	
Normaliseerimine	MAPE	MAPE	1	4	0,07	3,33	50,29
	MAPE	RMSE	1	5	0,13	3,30	49,90
	RMSE	MAPE	1,75	1	0,03	3,31	50,17
	RMSE	RMSE	1,75	5	0,13	3,31	50,07
Skaleerimine lõiku [-0,5,...,0,5]	MAPE/RMSE	MAPE	0,1	5	0,05	3,36	50,60
	MAPE/RMSE	RMSE	0,1	3	0,05	3,35	50,53

Tabel 2: Andmete eeltöötlemise meetodite võrdlus erinevate headusemõõdikute alusel mudeli prognoosivõimet hinnates

Juhul, kui prognoosivigade standardhälbe hindamiseks oleks kasutatud polünoomi asemel varasemat lineaarset mudelit, oleks epsiloni väärtus valemi 17 järgi tulnud 0,0106 varasema 0,0231 asemel. Pärast C ja epsiloni leidmist valemitega 16 ja 17 ning fikseerimist oleks parima keskmise suhtelise vea järgi valides parameeter sigma saanud väärtuse $\sigma = 0{,}31$. Parameetrite kombinatsioonide abil oleks osutunud optimaalseimaks $(\varepsilon; C)$ paariks $(\varepsilon; C) = (0{,}05; 0{,}3)$, mis oleks viinud keskmise suhtelise vea hinnanguni 0,0337 perioodil jaanuar-veebruar ning esimese nädalata 0,0287. Keskmine ruutviga testandmestikul oli 50,739.

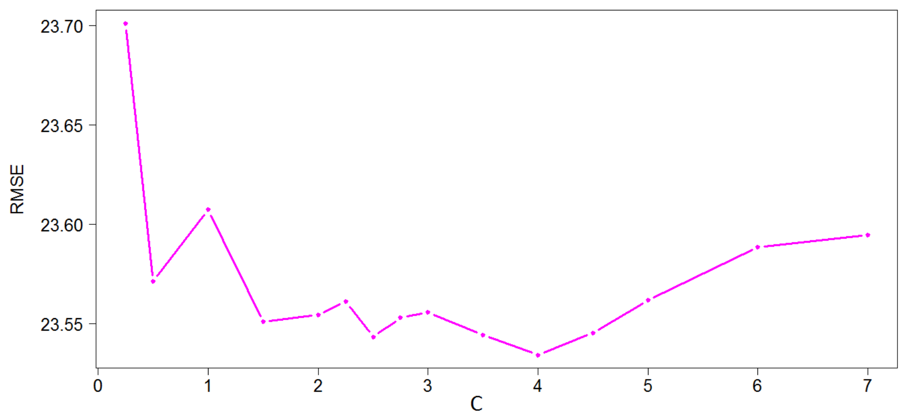
Võib öelda, elektritarbimise aegrida ei ole skaleerimisele tundlik ning keskmise suhtelise vea ning ruutvea järgi leitud parameetrite kombinatsioonid viivad sarnaste tulemusteni.

4.6.2 Analüütilisi lähendeid leidmata

Esmalt normaliseeritakse valemi 20 järgi andmed ning seejärel alustatakse sobitamist aegridadele spetsialiseeritud valideerimismeetodi abil. Esmalt kasutatakse käsku *trainControl* erinevate argumentide korral, kuid kuna see käsk on ühesammuline (treeningvalim kas kasvab või nihkub korraga ühe sammu) ning tegemist on tunnitäpsuste andmetega kahe aasta kohta, ei ole antud lähenemine piisavalt tõhus ning on liiga ajamahukas.

Programmi efektiivsemaks muutmiseks, antakse mudelile treenimiseks kindlaksmääratud valimid. Igast kuust aastal 2014 valitakse kaks päeva: üks nädalasisene ning teine nädalavahetuse päev, mis on testvalimiteks. Treenimiseks on alati kõik vastavale päevale eelnenud tunnid alates 2013. aasta algusest.

Parimaks C-ks vaikimisi epsiloni väärtuse $\varepsilon = 0{,}1$ korral (joonis 15) on 4. Antud mudeli keskmine suhteline viga üle kogu perioodi on 0,0328 ning esimese nädalata 0,0277. Keskmine ruutviga on 49,97. Mudel kasutab 5734 tugivektorit. Epsiloni väärtuse $\varepsilon = 0,02$ korral on esialgsest kolmest C variandist parim $C = 1$ ning σ hinnanguks tuleb $\sigma = 4,258$. Väiksema epsiloni väärtuse korral kasutab mudel 14403 tugivektorit. Antud mudeli ruutviga on 49,95. Keskmised suhtelised vead üle kogu perioodi ning esimese nädalata on, ümardades nelja kohani pärast koma, võrdsed.



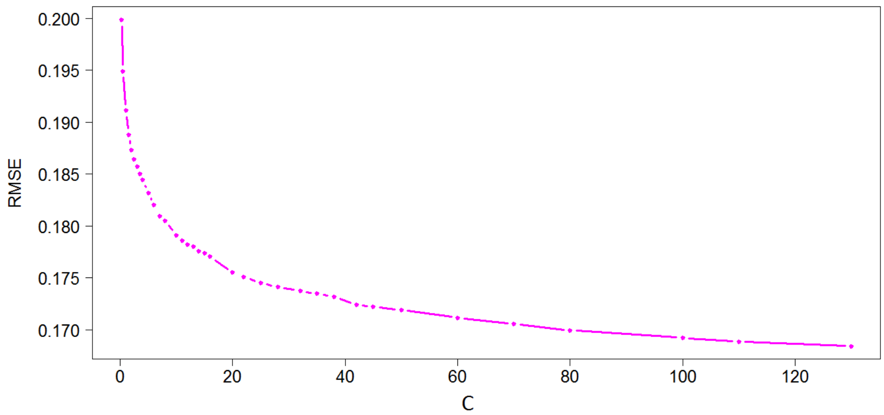
Joonis 15: Keskmise ruutvea sõltuvus C-st fikseeritud sigma korral

4.7 Kolmas tugivektorregressiooni abil leitud mudel

Kolmas tugivektorregerssiooni mudel on lineaarse mudeli 3 analoog. Kasutusel on argumendid 24 tundi tagasi olnud tarbimise kohta ning tunni ja päeva koosmõju. Mudeli parameetrite hindamisel kasutatakse isedefineeritud treening- ning testvalimeid sarnaselt teisele, aegridadele vastavat valideerimisemeetodit kasutavatale tugivektorregressiooni mudelile.

Vaikimisi parameetri epsilon väärtuse $\varepsilon = 0,1$ ning C väärtuste 0,25, 0,5 ja 1 korral hinnatakse sigma väärtuseks $\sigma = 0,3635$. Parim parameetrite kombinatsioon on C väärtuse $C = 1$ korral. Tulemust proovitakse parandada, laiendades otsitavate parameetrite väärtuste hulka. Parameetrile C antakse väärtused vahemikus $(0, 25, \ldots, 130)$

Jooniselt 16 on näha, kuidas parameetri C kasvades esialgu keskmine ruutviga kahaneb küllaltki kiiresti 0,2-lt 0,175-ni, mis on normaliseeritud andmete korral tõlgendatav 2,5 protsendina. Edasi muutub RMSE peaaegu konstantseks. Arvutusaeg läheb aga C kasvades suuremaks ning C väärtuste 60 ja 100 korral erineb mudelite treeningviga 0,000305. Seega valitakse sobivaks parameetrite kombinatsiooniks $\varepsilon = 0,1$, $\sigma = 0,3635$ ning $C = 60$. Mudel kasutab 6834 tugivektorit. Mudeli keskmine suhteline viga arvutatuna üle jaanuari ning veebruari on 0,0302 ja ilma esimese nädalata 0,0246. Keskmine ruutviga on 46,56.



Joonis 16: Keskmise ruutvea sõltuvus C-st fikseeritud sigma korral

Analüütilisteks lähenditeks saadakse valemi 16 järgi parameetri C jaoks 3 ning valemi 17 järgi, kasutades eelnevalt tehtud lineaarset mudelit prognoosijääkide standardhälbe hindamiseks, parameetri epsilon väärtuseks $\varepsilon = 0,0261$. Fikseeritud parameetrite $C = 3$ ja $\varepsilon = 0,0261$ korral on optimaalseim σ väärtus mudeli keskmise suhtelise vea järgi valideerimisandmestikul 1,42. Sellise parameetrite kombinatsiooniga sobitatud mudeli keskmine suhteline viga jaanuaris-veebruaris on 0,0296, ilma esimese nädalata 0,024. Keskmine ruutviga on 45,90.

5 Mudelite parandamine ja parima mudeli valik

Prognoosivea vähendamise eesmärgil muudetakse suhteid kasutavates mudelites (mudel 2 nii lineaarse regressiooni kui tugivektorregressiooni korral) suhete arvutamist. Tunnisuhted arvutati varem nädal tagasi toimunud ühe tunni tarbimise osakaaluna ööpäevasesse tarbimisse. Uue lähenemise kohaselt arvutatakse tunnisuhe eelmiste sama tüüpi päevade tunnisuhete abil, kasutades eksponentsiaalse silumise omadust $y_t = \alpha z_t + (1-\alpha)y_{t-1}$, koefitsiendiga $\alpha = 0,35$, kus z_t on esialgsete tunnisuhete rida [7, lk 11]. Päeva tüüp on kas tööpäev (esmaspäev-reede) või puhkepäev (nädalavahetused ja riigipühad). Kirjeldatud viisil uuendatud tunnisuhetega tehakse nii lineaarse regressiooni kui tugivektorregressiooni mudelid. Tabelis 3 on toodud uusi suhteid kasutanud mudelite parameetrid ning headusemõõdikute väärtused. Ükski kolmest testitud mudelist ei edesta ülejäänud kahte oluliselt.

	C	σ	ϵ	MAPE Jaan-veebr	MAPE nädalata	RMSE	Tugi-vektoreid
LINR	-	-	-	0,033	0,028	45,13	-
TVR	3	4,088	0,1	0,030	0,0259	43.92	7975
TVR $\varepsilon = 0,02$	1	4,088	0,02	0,031	0,0263	44,17	15256

Tabel 3: Uusi suhteid kasutavate mudelite parameetrid ning headusemõõdikute väärtused

Olenemata sellest, et jaanuari esimese nädala jaoks on suhete arvutamiseks kasutatud detsembri lõpu nädalavahetusi, suurendab esimese nädala prognoos keskmist suhtelist viga märkimisväärselt. Varasema suhete arvutamisega võrreldes on tehtud eeldus, et nädalasisesed ning -vahetuse päevad on omavahel sarnased ning ei kasutata eraldi nädalapäeva mõju, mis viib keskmise suhtelise vea vähenemiseni 0,21 protsenti.

Kuna eelnevalt koosmõjuga ning suhetega mudeli korral parameetri epsiloni vaikeväärtuse $\varepsilon = 0,1$ muutmisel väiksemaks kasvas tugivektorite arv väga suureks (kaasatud olid peaaegu kõik vaatlused) ning see ei andnud märkimisväärselt paremaid prognoose, sobitatakse edaspidi antud peatükis mudeleid erinevate parameetrite σ ning C paaride korral ja epsilon on fikseeritud $\varepsilon = 0,1$.

5.0.1 Lokaalse trendi kohta info lisamine

Ühelegi mudelile ei ole antud informatsiooni tarbimise koguse muutumise kohta lähiminevikus, mis sõltub ilmastikust- kas vaheldumas on aastaaeg soojemast külmemaks, valgemast pimedamaks või vastupidi. Mudelile info andmiseks lokaalse trendi kohta on mitu võimalust. Üheks neist on lisada tunnus kuu kohta. Lineaarse mudeli korral, kus on argumenttunnusteks 24h tagasi olnud tarbimine ning päeva ja tunni koosmõju, lisatakse koosmõjusse kuu. Tugivektorregressiooni jaoks on kõik tunnused aga arvulised ning kuu lisamine võib põhjustada kuude üleminekutel eksimusi. Kuude väärtused 1 ja 12 on mudeli jaoks üksteisest kauged, kuid aegrea mõttes on jaanuarile eelnev kuu detsember ning need vaatlused peaksid olema lähedased.

Teiseks võimalikuks modifikatsiooniks lokaalse trendi arvestamiseks on tunnuse 'aeg' lisamine, mis kujutab endast tundide loendurit. Ajatunnuse lisamisega saadakse kehvem tulemus kui uute suhetega tugivektorregressiooni korral, kuid parem kui esialgsete suhetega sobitatud tugivektorregressiooni puhul. Kolmandanda lisatakse temperatuur lineaarse funktsioonina. Viimasena antakse mudelile infot pikema tarbimise ajaloo kohta, lisades tunnuse nädal varem asetleidnud tarbimise kohta.

Tabelis 4 on toodud keskmised suhtelised vead protsentides kõikide suhteid kasutanud mudelite kohta. Lisaks on tabelis keskmised ruutvead. LINR tähistab lineaarset regressiooni, ülejäänud veerud esindavad tugivektorregressiooni mudeleid.

	LINR	esialgsed	uued LINR	uued	uued + nädal LINR	uued + aeg	uued + temp	uued + nädal
MAPE Jaan-veebr	3,37	3,28	3,27	3,07	3,07	3,17	4,17	2,87
MAPE nädalata	2,86	2,77	2,78	2,59	2,63	2,92	3,54	2,59
RMSE	50,23	49,95	45,132	43,92	41,95	44,15	64,56	42,39

Tabel 4: Suhetega mudelite keskmised suhtelised vead

Mudelid, mis kasutavad eelmiste sama tüüpi päevade pealt arvutatud suhteid, on natuke täpsemad kui nädalataguse tarbimise pealt arvutatud suhetega mudelid. Kõige väiksema keskmise suhtelise veaga mudel on tugivektorregressiooni mudel, kus suhted on arvutatud sama tüüpi päevade pealt ning lisatud on nädalatagune tarbimine. Keskmise ruutvea järgi on parim lineaarne regressiooni mudel, kuhu on lisatud nädalatagune tarbimine.

Tabelis 5 on ära toodud keskmised suhtelised vead protsentides ning keskmised ruutvead kõikide koosmõjuga mudelite kohta. Parimad koosmõjuga mudelid on lineaarse regressiooni mudel, kus on argumentideks tunni ja päeva koosmõju ning nädalatagune tarbimine ja samade argumenttunnustega tugivektorregressiooni mudel.

	päev *tund LINR	päev *tund	päev *tund *kuu LINR	päev *tund *kuu	päev *tund + aeg	päev *tund + temp	päev *tund + nädal	päev *tund + nädal LINR
MAPE Jaan-veebr	3,01	2,96	3,11	3,08	3,61	5,94	2,60	2,89
MAPE nädalata	2,43	2,40	2,66	2,64	3,18	5,12	2,23	2,32
RMSE	46,90	45,90	46,92	46,93	52,95	87,04	41,68	43,41

Tabel 5: Koosmõjuga mudelite keskmised suhtelised vead

Tabelis 6 on toodud keskmine suhteline viga protsentides aegrea mudeli ning tugivektorregressiooni mudeli 1 kohta.

	ARIMA	TVR mudel 1
MAPE Jaan-veebr	2,84	2,22
MAPE nädalata	2,27	1,88
RMSE	45,99	34,27

Tabel 6: Aegrea ning esimese tugivektorregressiooni mudeli keskmised suhtelised vead

Parima mudeli valimisel kaalutakse viie variandi vahel:

- ARIMA mudel
- Esimene tugivektorregressiooni mudel artikli [1] põhjal
- päeva ja tunni koosmõjuga lineaarne regressioon, kuhu on lisatud nädalatagune tarbimine
- korrigeeritud suhetega leitud tugivektorregressiooni mudel, kuhu on lisatud nädalatagune tarbimine
- päeva ja tunni koosmõjuga tugivektorregressiooni mudel, kuhu on lisatud nädalatagune tarbimine.

Eespool mainitud mudeleid kasutatakse märtsi ja aprilli prognoosimisel, lisades testperioodile jaanuari ning veebruari andmed ning võrreldes keskmist suhtelist ja ruutviga sellel perioodil. Tulemused on toodud tabelis 7. Ainuke mudel, mille korral märtis-aprillis vead kasvavad, on esimene tugivektorregressiooni mudel. Selle konkreetse mudeli puhul vähendati ka parameetri ε väärtust, kuna selline tegevus andis varasemalt väiksemaid pronoosivigu, kuid seekord vähenes viga minimaalselt. Lisaks vaadeldi märtsi ja aprilli prognoosimist peatükis 4.5 valituks osutnud mudeli abil, millega saadi keskmine suhteline viga 3,5 protsenti ning ruutviga 51,70. Seega uute andmete lisamiseega ning epsiloni vähendamisega saadi uuel perioodil oluliselt kehvem tulemus kui eelmisel perioodil ning edaspidises analüüsis seda mudelit ei käsitleta. Tabelis 7 on toodud viie valituks osutunud mudeli keskmine suhteline viga ning keskmine ruutivga perioodil märts-aprill 2015.

	ARIMA	uued suhted + nädal	päev*tund + nädal	päev*tund + nädal LINR	TVR 1
MAPE	2,03	2,50	2,35	2,31	2,94
RMSE	30,57	35,09	33,96	32,95	39,09

Tabel 7: Parimate mudelite keskmised ruutvead ning keskmised suhtelised vead protsentides märtsi ning aprilli tarbimise prognoosimisel

Tabelis 7 toodud mudelite märtsi ja aprilli tarbimise prognoose võrreldakse lähemalt. Esmalt vaadeldakse tegeliku ja prognoositud väärtuste vahesid ning absoluutseid vahesid. Kõik keskmised eksimused on negatiivse märgiga, mis viitab sellele, et mudeid kipuvad tarbimist ülehindama.

	ARIMA	uued suhted + nädal	päev*tund + nädal	päev*tund + nädal LINR
Keskmine eksimus MWh	-3,43	-4,09	-4,69	-3,42
Keskmine absoluutne eksimus MWh	19,38	23,38	22,18	21,63

Tabel 8: Tegeliku ja prognoositud tarbimise ning asboluutse tegeliku ja prognoositud tarbimise vahede keskmised

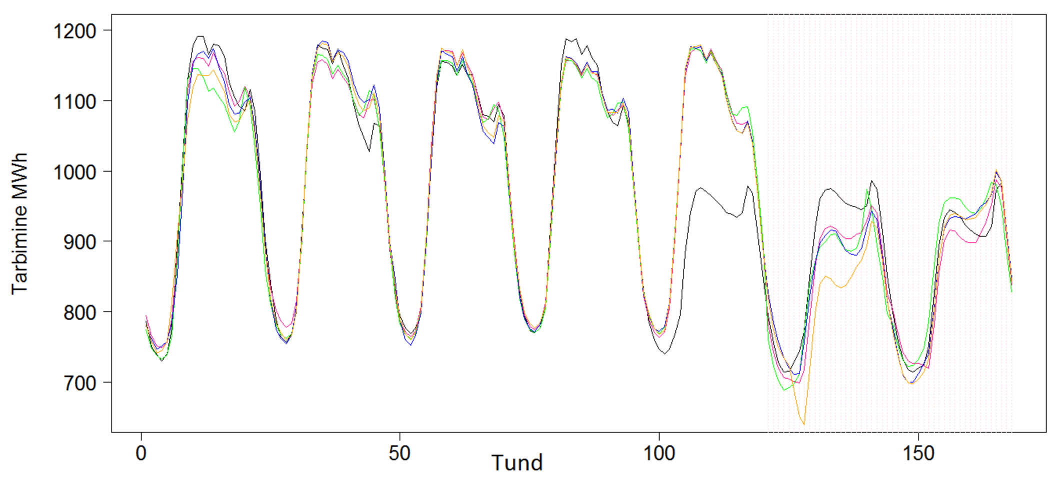
Tabelis 9 on toodud keskmine suhteline viga märtis ning aprillis töö- ning puhkepäevade lõikes.

	ARIMA	uued suhted + nädal	päev*tund + nädal	päev*tund + nädal LINR
MAPE nädalavahetustel	2,31	3,40	2,61	2,79
MAPE tööpäevadel	1,91	2,13	2,21	2,09

Tabel 9: Keskmised suhtelised vead nädalasisestel päevadel ning nädalavahetustel

Perioodi märts-aprill langeb kaks riigipüha. Suur reede 3. aprillil ning ülestõusmispühade esimene püha 5. aprillil. Teine riigipüha on pühapäevasel päeval ning seega ei erine tarbimine oluliselt tavapärasest. Suurel reedel aga eksivad kõik mudelid keskmiselt rohkem kui 13 protsenti. Nädalasisesed riigipühad on suurimad veaallikad ning toimub tarbimise koguse ülehindamine, mis tekitab vajaduse luua riigipühadega tegelemiseks paremad meetmed.

Joonisel 17 on toodud 30.03-05.04 2015 elektritarbimine (must) koos ARIMA prognoosiga (roosa), suhetega mudeli prognoosiga (roheline), koosmõjuga tugivektorregressiooni mudeli prognoosiga (sinine) ning koosmõjuga lineaarse regressiooni mudeli prognoosiga (oranž). Suurel reedel, 3. aprillil, on kõik mudelid tarbimist oluliselt üle hinnanud.



Joonis 17: Perioodi 30.03-05.04 tarbimine ja prognoosid

Uute suhete arvutamine eksponentsiaalse silumise teel tagas väljaspool nädalavahetust riigipühadele ehk suurimatele veaallikatele õigema kõvera kuju. Suhetega mudeli korral oli riigipühade tarbimise hulga probleemi kõrvaldamise üheks lahenduseks lisada reede ning laupäeva ülemineku indikaatorile väärtused 1 ka nädalasiseste riigipühade korral. Samuti lisada pühapäeva ning esmaspäeva ülemineku indikaatorile väärtus 1 neile päevadele, mis järgnevad nädalasisestele riigipühadele (ning on ka ise nädalasisesed päevad). Mudel saab kuju, kus

$$\text{tarbimine}_i = (c_1\text{eelneva 24h tarbimine} + c_2 I_{pt} + c_3 I_{tp} + c_4) * C_h + \varepsilon_i$$

Indikaator I_{pt} on defineeritud

$$I_{pt} = \begin{cases} 1, & \text{kui toimub üleminek puhkepäevalt tööpäevale} \\ 0, & \text{muidu} \end{cases}$$

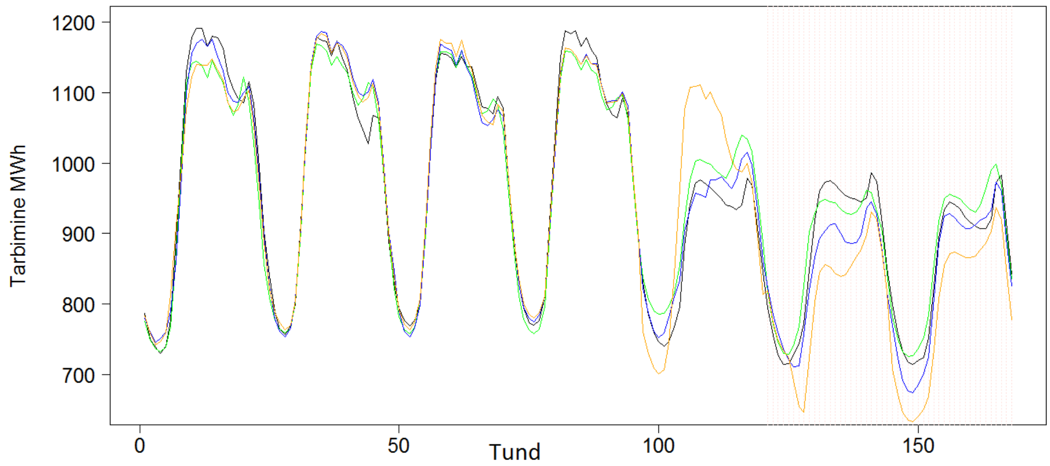
Sarnaselt on defineeritud I_{tp}, mis tähistab üleminekut tööpäeva ning puhkepäeva vahel. Tööpäevade all peetakse silmas päevi esmaspäevast reedeni. Puhkepäevaks loetakse kõik nädalavahetuse päevad koos riigipühadega.

Koosmõjuga lineaarse regressiooni ning tugivektorregressiooni mudeli korral lisatakse riigipühade indikaator. Tabelis 10 on toodud rigipühasid silmas pidades modifiteeritud mudelite keskmised suhtelised vead ning keskmine ruutviga märtsis ning aprillis. Saadud tulemused on samas suurusjärgus ARIMA mudeli vigadega, päeva ja tunni koosmõju ning nädalataguse tarbimisega tugivektorregressiooni mudel edestab ARIMA suhtelist viga ligikaudu 0,1 protsendiga.

	uued suhted + nädal	päev*tund + nädal	päev*tund + nädal LINR
MAPE	2,33	2,20	2,37
RMSE	29,53	28,44	31,12

Tabel 10: Keskmine suhteline viga ning keskmine ruutviga märtsi ning aprilli tarbimise prognoosimisel, arvestades riigipühi

Jooniselt 18 on näha, et prognoos on riigipühal (Suurel reedel) oluliselt paranenud. Kõige rohkem on probleeme riigipühal erineva tarbimise käitumise tabamisel lineaarse regressiooni mudelil.



Joonis 18: Perioodi 30.03-05.04 tarbimine ja prognoosid

Mudelitega, kus arvestati ka riigipühasid prognoositi võrdluse mõttes jaanuar ning veebruar, et uurida mudeli käitumist suurema arvu riigipühade korral. Tulemused on tabelis 11. Tabelist on näha, et isegi riigipühade arvestamisega on jaanuari esimene nädal suureks veaallikaks. Võrreldes tabelites 4 ja 5 toodud tulemustega paistab silma, et lineaarse mudeli korral paranes keskmine suhteline viga ligikaudu 0,2 protsenti ja koosmõjuga tugivektori mudeli korral ligikaudu 0,15 protsenti, kuid suhetega mudeli korral oli parandus väiksem. Keskmine ruutviga vähenes kõigil kolmel juhul, kõige rohkem koosmõjuga tugivektormudeli korralligikaudu 5 võrra.

	uued suhted + nädal	päev*tund + nädal	päev*tund + nädal LINR
MAPE	2,83	2,44	2,69
MAPE nädalata	2,44	2,24	2,24
RMSE	39,37	34,75	39,39

Tabel 11: Keskmine suhteline viga ning keskmine ruutviga jaanuari ning veebruari tarbimise prognoosimisel, arvestades riigipühi

6 Kokkuvõte

Käesoleva magistritöö eesmärgiks on ennustada lühiajaliselt Eesti elektritarbimist, kasutades elektritarbimise ajalugu ning ilmaandmeid. Esmalt antakse ülevaade tugivektorregressiooni teooriast ning tehakse meeldetuletus aegridade kohta. Lisaks tutvustatakse tarkvara R paketti *Caret* ning selle funktsioone. Räägitakse aegrealiste andmete eripärast, tugivektorregressiooni parameetrite valikust ning andmete eeltöötlemisest.

Praktilises osas alustatakse võrdlusbaasi koostamisega lineaarse regressiooni mudelite näol. Nendes mudelites kasutatakse tarbimise ajaloona 24 h varem asetleidnud tarbimist ning päeva ja kellaaja koosmõju või viimase ööpäeva tarbimist koos nädalavahetuse ja nädalapäevade indikaatoritega, korrutatuna tunni tarbimise suhtega ööpäevasesse tarbimisse. Võrdluseks testitakse ka ühte ARIMA mudelit. Hiljem tehakse lineaarse regressiooni mudelitega analoogsed mudelid tugivektorregressiooni abil.

Mudelite võrdlemiseks kasutatakse keskmist suhtelist viga ja keskmist ruutviga. Saadud koondatakse tulemused koondatakse tabelitesse ning valitakse välja parimad mudelid, mille korral parandatakse prognoositäpsust, lisades ning muutes tarbimise ajalugu kirjeldavaid tunnuseid ja kasutades riigipühadega arvestavaid meetmeid. Parimateks mudeliteks osutuvad ARIMA mudel, korrigeeritud suhete arvutamisega ning päeva ja tunni koosmõjuga tugivektorregressiooni mudelid, kuhu on lisatud nädalatagune tarbimine ja viimasega analoogne lineaarse regressiooni mudel.

Töö lõpus on lisad, kus on toodud kasutatud tarkvara peamine kood. Andmekandjal on lisatud kasutatavad koodid koos andmestikega. Aeganõudvama analüüsi korral on salvesatud ka tulemused mudelite ning prognooside näol.

A Lineaarsete mudelite koodid

A.1 Mudeli 1 kood

```
# asukoht
setwd("C:/Users/Lemz/Documents/Cliona␣kraam/Kool/Magistritöö")
# treeningandmestik
yl1.1 = read.table("prognoosiks2.csv", header = T, sep = ",", dec = ".")
yl1.1$X = NULL #eelmine-24h tagasi olnud tarbimine
attach(yl1.1) #eelmise_tarbimine-eelmise h tarbimine,tarbimine24-eelneva öp tarbimine
# konkreetse mudeli jaoks väiksem andmestik
andmed = data.frame(kuupaevad, tunnid, tarbimine, eelmise_tarbimine, temperatuur, kuu)
attach(andmed)
# mudel
mudel = lm(tarbimine~eelmise_tarbimine+factor(tunnid),data=andmed)
# PROGNOOSIME jaanuari-veebruari 2015
uusdata = read.table("jaanuarveebruar_mudel1.csv", header = T,sep = ",", dec = ".")
tundide_arv = (31+28)*24
prognoos = rep(NA,tundide_arv)
prognoos = predict(mudel, data.frame(eelmise_tarbimine = uusdata$eelmise_tarbimine,
    tunnid = uusdata$tunnid))
# koos temperatuuriga
mudel1=lm(tarbimine ~ eelmise_tarbimine+factor(tunnid)+temperatuur,data = andmed)
uusdata1=read.table("jaanuarveebruar_mudel1_temp.csv",header=T,sep =",",dec=".")
# prognoos
prognoos1 = rep(NA,tundide_arv)
prognoos1 = predict(mudel1, data.frame(eelmise_tarbimine=uusdata1$eelmise_tarbimine,
    tunnid = uusdata1$tunnid, temperatuur = uusdata1$t_jv))
# MUDELI SOORITUS
tegelik = read.table("jaanuarveebruar.csv", header = T,sep = ";", dec = ",")$Tarbimine
# teeme graafiku võrreldes reaalse jaanuariga, prognoos = prognoos või prognoos=
    prognoos1
par(mar=c(4, 4, 4, 2) )
x = seq(1,tundide_arv)
plot(x,tegelik, type = "l")
lines(prognoos, col = "deeppink")
# keskmine suhteline viga ja keskmine ruutviga
mape = mean((abs(tegelik-prognoos))/tegelik)
rmse = sqrt((1/length(prognoos))*sum((tegelik-prognoos)**2))
# eksimused
analyys = data.frame(uusdata, prognoos) # uusdata1
vahe = analyys$tarbimine - analyys$prognoos # mean(vahe)
# max/min eksimus ja aeg
analyys[which(abs(vahe) == max(abs(vahe))),]
```

A.2 Mudeli 2 kood

```
# asukoht
setwd("C:/Users/Lemz/Documents/Cliona␣kraam/Kool/Magistritöö")
# treeningandmestik
andmed = read.table("mudel2_treening.csv", header = T, sep = ",", dec = ".")
# tegelik tarbimine
jaanuarveebruar = read.table("jaanuarveebruar1.csv", header = T,sep = ";", dec = ",")
# testandmestik
uusdata = read.table("mudel2_test.csv", header = T, sep = ",", dec = ".")
# mudel
mudel = lm(tarbimine ~ korrutis2 + korrutisE + korrutisL, data = andmed)
# prognoosime jaanuari 2015
tundide_arv = (31+28)*24
prognoos = rep(NA,tundide_arv)
prognoos = predict(mudel, data.frame(korrutisE = uusdata$korrutisE,korrutisL = uusdata$
    korrutisL,
```

```
korrutis2 = uusdata$korrutis2))
kuupaev = rep(seq(as.Date("2015-01-01"), as.Date("2015-02-28"), by="days"), each = 24)
paevad = weekdays(as.Date(kuupaev))
# keskmine suhteline viga ning näfalata keskmine suhteline viga, 24. veebuar on
    [1297:1320]
mape = mean((abs(jaanuarveebruar$Tarbimine-prognoos))/jaanuarveebruar$Tarbimine)
mape = mean((abs(jaanuarveebruar$Tarbimine[169:length(prognoos)]-prognoos[169:length(
    prognoos)]))/jaanuarveebruar$Tarbimine[169:length(prognoos)])
# keskmine ruutviga
rmse = sqrt((1/length(prognoos))*sum((jaanuarveebruar$Tarbimine-prognoos)**2))
# JOONIS 26.01-01.02
par(mar=c(5, 5, 4, 3) )
x = seq(1:168)
plot(x,jaanuarveebruar$Tarbimine[601:768],type = "l", ylab="Tarbimine (MWh)",xlab="Tund"
    ,cex.lab=1.5,        cex.axis=1.6,mgp=c(3.8,1,0),las = 1)
abline(v = x[paevad[601:768] == "laupäev"], col = "pink", lty = 3)
abline(v = x[paevad[601:768] == "pühapäev"],col = "pink", lty = 3)
lines(prognoos[601:768], col = "deeppink")
# Joonis 02.02-08.02 korral indeksid [769:936]
```

A.3 Mudeli 3 kood

```
setwd("C:/Users/Lemz/Documents/Cliona kraam/Kool/Magistritöö")
# treening ja test
andmed = read.table("mudel3_treening.csv", header = T, sep = ",", dec = ".")
uusdata = read.table("mudel3_test.csv", header = T, sep = ",", dec = ".")
# tegelik, sisaldub ka uusdata 1. veerus
jaanuarveebruar = read.table("jaanuarveebruar1.csv", header = T,sep = ";", dec = ",")
# mudel
mudel = lm(tarbimine ~ eelmine + factor(tunnid)*factor(paev)   , data = andmed)
tundide_arv = (31+28)*24
prognoos = rep(NA,tundide_arv)
prognoos= predict(mudel,data.frame(eelmine=uusdata$eelmine,tunnid=uusdata$tunnid,paev=
    uusdata$paev))
# jääkide graafik
par(mar=c(5, 5, 4, 3) )
par(mfrow=c(1,2))
residuals = jaanuarveebruar$Tarbimine - prognoos
fitted = prognoos - residuals
plot(fitted,residuals,pch = 19,xlab ="Sobitatud väärtused",ylab="Jäägid",ylim=c
    (-200,200),
cex.lab = 1.6,cex.axis=1.6,mgp=c(3.6,1,0),xlim=
c(540,1500),las = 1)
lines(fitted, rep(0,length(fitted)), col = "red")
# esimene nädal ja 24.02,25.02 välja
ilma_tarb = c(jaanuarveebruar$Tarbimine[169:1296],jaanuarveebruar$Tarbimine[1344:1416])
ilma_prognoos =c(prognoos[169:1296],prognoos[1344:1416])
residuals1 = ilma_tarb - ilma_prognoos
fitted1 = ilma_prognoos - residuals1
plot(fitted1,residuals1,ylim =c(-200,200),pch =19,xlab ="Sobitatud väärtused",ylab="Jää
    gid",cex.lab=1.6, cex.axis=1.6,mgp=c(3.6,1,0),xlim=c(540,1500),las=1)
lines(fitted, rep(0,length(fitted)), col = "red")
# nädala graafik
par(mfrow=c(1,1))
kuupaev = rep(seq(as.Date("2015-01-01"), as.Date("2015-02-28"), by="days"), each = 24)
paevad = weekdays(as.Date(kuupaev))
x = seq(1:168)
plot(x,jaanuarveebruar$Tarbimine[769:936],type = "l", ylab="Tarbimine (MWh)",xlab="Tund"
    ,cex.lab=1.5,cex.axis =1.6,mgp=c(3.8,1,0),las=1)
abline(v = x[paevad[769:936] == "laupäev"], col = "pink", lty = 3)
abline(v = x[paevad[769:936] == "pühapäev"],col = "pink", lty = 3)
lines(prognoos[769:936], col = "deeppink")
# keskmine suhteline viga nädalata ja kogu periood
mape = mean((abs(jaanuarveebruar$Tarbimine[169:length(prognoos)]-prognoos[169:length(
    prognoos)]))
```

```
/jaanuarveebruar$Tarbimine[169:length(prognoos)])
mape = mean((abs(jaanuarveebruar$Tarbimine−prognoos))/jaanuarveebruar$Tarbimine)
# ilma 24. veebruarita ja esimese nädalata
mape = mean((abs(c(jaanuarveebruar$Tarbimine[169:1296],jaanuarveebruar$Tarbimine
    [1321:1416])−c(prognoos[169:1296],prognoos[1321:1416])))
/c(jaanuarveebruar$Tarbimine[169:1296],jaanuarveebruar$Tarbimine[1321:1416]))
# keskmine ruutviga
rmse=sqrt((1/length(prognoos))*sum((jaanuarveebruar$Tarbimine−prognoos)**2))
# eksimuste uurimine
analyys = data.frame(uusdata, prognoos)
analyys$vahe = analyys$tarbimine − analyys$prognoos
mean(analyys$vahe) # 5.743159
analyys[which(abs(vahe) == max(abs(vahe))),] # 2015−01−01 tund 9  269.1399
analyys[which(abs(vahe) == min(abs(vahe))),] # 2015−02−07 tund 21 0.02810136
tegelik_kesk = aggregate(analyys$tarbimine ~ tunnid , analyys, mean)[,2]
vahe_kesk = aggregate(abs(analyys$vahe) ~ tunnid , analyys, mean)[,2]
protsent= (vahe_kesk/tegelik_kesk)*100
plot(protsent,type="l",ylab="Keskmine eksimise protsent",xlab = "Tund",col=14,ylim=c
    (1,7),las=1)
min(abs(vahe_kesk))
```

B Aegrea kood

```
# asukoht
setwd("C:/Users/Lemz/Documents/Cliona kraam/Kool/Magistritöö")
joonised = function(rida, mitu ){
  layout(1:3)
  plot(rida, type = "l")
  acf(rida, mitu)
  pacf(rida, mitu)
  layout(1)
}
# tegelik
jaanuarveebruar = read.table("jaanuarveebruar1.csv", header = T,sep = ";", dec = ",")
# treeningrida
andmed1 = read.table("prognoosiks2.csv", header = T, sep = ",", dec = ".")$tarbimine
# aegread
rida = ts(andmed1, start = 2013, frequency = 24*365)
rida_test = ts(jaanuarveebruar$Tarbimine, start = 2015, frequency = 24*365)
joonised(rida,165) # ei ole statsionaarne
PP.test(rida)
# diferentsimine
diff_rida = diff(rida,168)# joonised(diff_rida,400)
# mudel
mudel = arima(rida, order = c(25,0,25), seasonal = list(order=c(1,1,1),period=7*24),
    method="CSS",
fixed=c(NA,rep(0,21),NA,NA,NA,NA,rep(0,21),NA,NA,NA,NA,NA))
paevade_arv = 31+28 # jaanuarveebruar
ennustus = rep(NA, paevade_arv*24)
tegelik = window(rida_test, start = 2015) # terve testperiood
hetk = 17520 # 2 aastat
# prognoosimine
for (i in seq(1,(paevade_arv*24),24)){
  mudel = arima(c(rida,tegelik)[1:(hetk)], seasonal=list(order=c(1,1,1),period=7*24),
      order=c(25,0,25),fixed =mudel$coef)
  ennustus[i:(i+23)] = predict(mudel, 24)$pred
  hetk = hetk + 24
}
# keskmine suhteline viga
mape = mean(abs(tegelik−ennustus)/tegelik)#  0.02839343
mape = mean((abs(jaanuarveebruar$Tarbimine[169:length(prognoos)]−prognoos[169:length(
    prognoos)]))/jaanuarveebruar$Tarbimine[169:length(prognoos)])
mape # 0.02270098
# keskmine ruutviga
```

```
rmse = sqrt((1/length(ennustus))*sum((jaanuarveebruar$Tarbimine-ennustus)**2))# 45.98618
kuupaev = rep(seq(as.Date("2015-01-01"), as.Date("2015-02-28"), by="days"), each = 24)
paevad = weekdays(as.Date(kuupaev))
# JOONIS
par(mar=c(5, 5, 4, 3) )
x = seq(1:168)
plot(x,jaanuarveebruar$Tarbimine[601:768],type = "l", ylab="Tarbimine␣(MW)",xlab="Tund",
    cex.lab=1.5,    cex.axis=1.6,mgp=c(3.8,1,0),las=1)
abline(v = x[paevad[601:768] == "laupäev"], col = "pink", lty = 3)
abline(v = x[paevad[601:768] == "pühapäev"],col = "pink", lty = 3)
lines(x,jaanuarveebruar$Tarbimine[601:768],  type = "l")
lines(prognoos[601:768], col = "deeppink")
# esimene nädal indeksid [1:168]
```

C Tugivektorregressiooni mudelite koodid

C.1 Tugivektorregressiooni mudeli 1 kood

```
# asukoht
setwd("C:/Users/Lemz/Documents/Cliona␣kraam/Kool/Magistritöö")
# paketti installeerimine
# install.packages("caret")
library("caret")
# treening
treening = read.table("24tagasi.csv", header = T, sep = ",", dec = ".")
treening$X = NULL
# test
test = read.table("24tagasi_test1.csv", header = T, sep = ",", dec = ".")
test$X = NULL
# tegelik
jaanuarveebruar = read.table("jaanuarveebruar1.csv", header = T,sep = ";", dec = ",")
# skaleerimine
preProcvalues = preProcess(treening, method =  "range")
treening_transf = predict(preProcvalues, treening)
test_transf3 = predict(preProcvalues, test)
test_transf3$tarbimine = NULL
# mudeli tegemine
# oma_telg = expand.grid(C = 3, sigma = c(seq(0.01,0.11,0.01),0.13,0.15,0.2)  )
# teeme mudeli
control = trainControl( method = "repeatedcv",number = 5,  repeats = 2, savePred = T)
set.seed(1500)
milos<- train(tarbimine~., data = treening_transf, method = "svmRadial",
            trControl = control) # epsilon = 0.02,tuneGrid = oma_telg
# Lisatud magistritööle, kaustas TVR1
# load("vaikeparameetritega_mudel.Rdata") # C = 1, epsilon = 0,01, mudeli nimi 'milos'
# load =("Milos_mudel_C3sigma008.Rdata") # C =3 ning sigma = 0,08 korral, nimi 'S_sigma'
# load("Miloš_uuseps1") # C = 1, eps = 0,02, nimi 'S1'
# load("Miloš_uuseps2") # C = 6, eps = 0,02, nimi 'S2'
# laadimise näide
mudel = load("Miloš_uuseps2")
S2$finalModel
# prognoosimine ühe tunni kaupa
prognoos = predict(S2, newdata = test_transf3)
# tagasiskaleerimine
miin = preProcvalues$range[1,2]
maks =  preProcvalues$range[2,2]
prognoos = prognoos*(maks-miin) + miin
# keskmine suhteline viga
mape = mean((abs(jaanuarveebruar$Tarbimine-prognoos))/jaanuarveebruar$Tarbimine)
# rekursiivsed tunnid
nimed=c("tunnid","tagasi_1","tagasi_2","tagasi_3",
"tagasi_4","tagasi_5","tagasi_6","tagasi_7","tagasi_8","tagasi_9",
```

```
"tagasi_10","tagasi_11","tagasi_12","tagasi_13","tagasi_14",        "tagasi_15","tagasi
    _16","tagasi_17","tagasi_18", "tagasi_19","tagasi_20","tagasi_21","tagasi_22", "
    tagasi_23","tagasi_24","paev")
prognoos1 = rep(NA, length(jaanuarveebruar$Tarbimine))
for (i in seq(1,length(jaanuarveebruar$Tarbimine),24)){
  prognoos1[i] = predict(S2,newdata = test_transf3[i,]) # mudel muuta vastavalt nimele
  k = 1
  for (k in 1:23){
    uus=data.frame(matrix(as.numeric(c(test_transf3$tunnid[i+k], rev(prognoos1[i:(i+k-1)
        ]),test_transf3[i,2:(26-k-1)], test_transf3$paev[i])),nrow = 1))
    colnames(uus) <- nimed
    prognoos1[i+k] = predict(S2,newdata = uus)
  }
}
# tagasi skaleerimine
prognoos1_unscale= prognoos1*(maks-miin) + miin
# headus
mape=mean((abs(jaanuarveebruar$Tarbimine-prognoos1_unscale))/jaanuarveebruar$Tarbimine)
mape = mean((abs(jaanuarveebruar$Tarbimine[169:length(prognoos1_unscale)]-prognoos1_
    unscale[169:length(prognoos1_unscale)]))/jaanuarveebruar$Tarbimine[169:length(
    prognoos1_unscale)])
rmse=sqrt((1/length(prognoos1_unscale))*sum((jaanuarveebruar$Tarbimine-prognoos1_unscale
    )**2))
# nädalate kaupa vaatamine
mape_vektor = NA
for (i in seq(1,1416,168)){
  mape = mean((abs(jaanuarveebruar$Tarbimine[i:(i+168)]-prognoos1_unscale[i:(i+168)]))
        /jaanuarveebruar$Tarbimine[i:(i+168)])
  mape_vektor = c(mape_vektor,mape)
}
(mape_vektor*100)[2:9]
mean(mape_vektor, na.rm = TRUE)
# graafik
kuupaev = rep(seq(as.Date("2015-01-01"), as.Date("2015-02-28"), by="days"), each = 24)
paev_tekst = weekdays(as.Date(kuupaev))
par(mar=c(5, 5, 4, 3) )
x = seq(1:168)
plot(x,jaanuarveebruar$Tarbimine[601:768],  type = "l", ylab = "Tarbimine␣(MW)", xlab =
    "Tund",cex.lab=1.5,cex.axis=1.6,mgp=c(3.8,1,0),las=1)
abline(v=x[paev_tekst[601:768]=="laupäev"],col="pink", lty=3)
abline(v=x[paev_tekst[601:768]=="pühapäev"],col="pink", lty=3)
lines(x,jaanuarveebruar$Tarbimine[601:768], type = "l")
lines(prognoos1_unscale[601:768], col = "deeppink")
```

C.2 Tugivektorregressiooni mudeli 2 kood

C.2.1 Analüütilise lähenemise kood

```
setwd("C:/Users/Lemz/Documents/Cliona␣kraam/Kool/Magistritöö")
library("caret")
jaanuarveebruar = read.table("jaanuarveebruar.csv", header = T,sep = ";", dec = ",")
# treening
treening = read.table("suhetega_treening.csv", header = T, sep = ",", dec = ".")
treening$X = NULL
# test
test = read.table("suhetega_test.csv", header = T, sep = ",", dec = ".")
test$X = NULL
# Jagame osadeks
trenn = treening[1:15313,]
valideerimine = treening[15313:17520,]
# normaliseerimine
preProcvalues = preProcess(trenn, method =  c("scale", "center"))
trenn_transf = predict(preProcvalues, trenn)
valideerimine_transf = predict(preProcvalues, valideerimine)
```

```
test_transf = predict(preProcvalues, test)
test_transf$tarbimine = NULL
valideerimine_transf$tarbimine = NULL
# lineaarne teisendus lõiku [-0.5, 0.5]
a = -0.5
b = 0.5
tarbimine = a + (trenn$tarbimine-min(treening$tarbimine))/(max(treening$tarbimine)-min(
    treening$tarbimine))*(b-a)
korrutisL = a + (trenn$korrutisL-min(treening$korrutisL))/(max(treening$korrutisL)-min(
    treening$korrutisL))*(b-a)
korrutisE = a + (trenn$korrutisE-min(treening$korrutisE))/(max(treening$korrutisE)-min(
    treening$korrutisE))*(b-a)
korrutis2 = a + (trenn$korrutis2-min(treening$korrutis2))/(max(treening$korrutis2)-min(
    treening$korrutis2))*(b-a)
trenn_transf = data.frame(tarbimine, korrutisE, korrutisL, korrutis2)
# peab ka siin kasutama treeningu parameetreid
tarbimine = a + (valideerimine$tarbimine-min(treening$tarbimine))/(max(treening$
    tarbimine)-min(treening$tarbimine))*(b-a)
korrutisL = a + (valideerimine$korrutisL-min(treening$korrutisL))/(max(treening$
    korrutisL)-min(treening$korrutisL))*(b-a)
korrutisE = a + (valideerimine$korrutisE-min(treening$korrutisE))/(max(treening$
    korrutisE)-min(treening$korrutisE))*(b-a)
korrutis2 = a + (valideerimine$korrutis2-min(treening$korrutis2))/(max(treening$
    korrutis2)-min(treening$korrutis2))*(b-a)
valideerimine_transf = data.frame(tarbimine, korrutisE, korrutisL, korrutis2)
korrutisL = a + (test$korrutisL-min(treening$korrutisL))/(max(treening$korrutisL)-min(
    treening$korrutisL))*(b-a)
korrutisE = a + (test$korrutisE-min(treening$korrutisE))/(max(treening$korrutisE)-min(
    treening$korrutisE))*(b-a)
korrutis2 = a + (test$korrutis2-min(treening$korrutis2))/(max(treening$korrutis2)-min(
    treening$korrutis2))*(b-a)
test_transf =data.frame(korrutisE, korrutisL, korrutis2)
# valemite rakendamine
y_kesk = mean(trenn_transf$tarbimine) # 0
sigma_y = sd(trenn_transf$tarbimine)
C = 3*sigma_y
# sigma hindamine
n = length(trenn$tarbimine)
tau = 3
mudel = lm(tarbimine ~ poly(korrutis2,10,raw = TRUE), data = trenn_transf)
d = 10
sigma = sqrt(n/(n-d)*1/n*sum((trenn_transf$tarbimine - predict(mudel))**2))
# epsiloni arvutamine
epsilon = tau*sigma*sqrt(log(n)/n)
tarb_kesk = mean(trenn$tarbimine)
tarb_sd = sd(trenn$tarbimine)
# sigma arvutamine fikseeritud C ja epsiloni korral
sigma_vektor = seq(0.01,3,0.03)
mape_vektor = rep(NA, length(sigma_vektor))
for (i in 1:length(sigma_vektor)){
  oma_telg = expand.grid(C = C,sigma=sigma_vektor[i] )
  control = trainControl( method = "none")
  set.seed(1500)
  SVMR <- train(tarbimine~., data = trenn_transf, method = "svmRadial", trControl =
      control,epsilon = epsilon, tuneGrid = oma_telg)
  prognoos = predict(SVMR, valideerimine_transf)
  prognoos = prognoos*tarb_sd + tarb_kesk
  # prognoos = (prognoos-a)/(b-a)*(max(treening$tarbimine)-min(treening$tarbimine)) +
      min(treening$tarbimine)
  mape = mean((abs(valideerimine$tarbimine-prognoos))/valideerimine$tarbimine)
  #rmse = sqrt(mean((valideerimine$tarbimine-prognoos)**2)) vastav valem kommenteerida
  mape_vektor[i] = mape
}
indeks = which(mape_vektor == min(mape_vektor))
# epsiloni ja C jaoks paarid
gridi_suurus = 35
C_vektor = rep(c(0.1,0.3,0.5,0.7,0.9,1,1.1), each = 5)
```

```
mape_vektor2 = rep(NA, length(C_vektor))
epsilon_vektor = rep(c(0.003,0.005,0.01, 0.05,0.09), 7)
for ( i in 1:gridi_suurus){
  oma_telg = expand.grid(C = C_vektor[i], sigma = sigma_vektor[indeks])
  control = trainControl( method = "none")
  set.seed(1500)
  SVMR <- train(tarbimine~., data= valideerimine_transf,method = "svmRadial", trControl
      = control,epsilon = epsilon_vektor[i],tuneGrid=oma_telg)
  prognoos = predict(SVMR, valideerimine_transf)
  prognoos = prognoos*tarb_sd + tarb_kesk
  # prognoos = (prognoos-a)/(b-a)*(max(treening$tarbimine)-min(treening$tarbimine))+min(
      treening$tarbimine)
  mape = mean((abs(valideerimine$tarbimine-prognoos))/valideerimine$tarbimine)
  #rmse = sqrt(mean((valideerimine$tarbimine-prognoos)**2)) vastav valem kommenteerida v
      älja
  mape_vektor2[i] = mape
}
indeks2 = which(mape_vektor2 == min(mape_vektor2))
# kogu treeningandmestikule sobitamine
valideerimine_transf = predict(preProcvalues, valideerimine)
kogu = rbind(trenn_transf, valideerimine_transf)
oma_telg = expand.grid(C = C_vektor[indeks2], sigma = sigma_vektor[indeks] )
control = trainControl( method = "none")
set.seed(1500)
SVMR1 <- train(tarbimine~., data = kogu, method = "svmRadial", trControl = control,
    tuneGrid = oma_telg, epsilon = 0.05)
prognoos = predict(SVMR1,test_transf)
# tagasiteisendus
prognoos = (prognoos-a)/(b-a)*(max(treening$tarbimine)-min(treening$tarbimine)) + min(
    treening$tarbimine)
#prognoos = prognoos*tarb_sd + tarb_kesk # normaliseerimise korral
# keskmine suhteline viga esimese nädalata
mape = mean((abs(jaanuarveebruar$Tarbimine[169:length(prognoos)]-prognoos[169:length(
    prognoos)]))/jaanuarveebruar$Tarbimine[169:length(prognoos)])
mape = mean((abs(jaanuarveebruar$Tarbimine-prognoos))/jaanuarveebruar$Tarbimine)
# ruutkeskmine viga
rmse = sqrt((1/length(prognoos))*sum((jaanuarveebruar$Tarbimine-prognoos)**2))# 46.89919
```

C.2.2 Analüütiliste lähenditeta

```
# asukoht
setwd("C:/Users/Lemz/Documents/Cliona␣kraam/Kool/Magistritöö")
# install.packages("caret")
library("caret")
# andmed
jaanuarveebruar = read.table("jaanuarveebruar1.csv", header = T,sep = ";", dec = ",")
treening = read.table("andmed_suhted_treening.csv", header = T, sep = ",", dec = ".")
treening$X = NULL
test = read.table("andmed_suhted_test.csv", header = T, sep = ",", dec = ".")
test$X = NULL
# skaleerimine
preProcvalues = preProcess(treening, method = c("scale", "center"))
treening_transf = predict(preProcvalues, treening)
test_transf = predict(preProcvalues, test)
test_transf$tarbimine = NULL
# timeslice ehk ajakulukas variant
myTimeControl <- trainControl(method= "timeslice",initialWindow = 4248,
                               horizon = 1416,fixedWindow = TRUE)
mudel <- train(tarbimine ~ .,data = treening_transf,
method = "svmRadial", trControl = myTimeControl)
# enda kindlaks määratud test ja treening
fit_on <-list(rs1=1:8952,rs2=1:9163,rs3=1:9696,rs4=1:10128,
          rs5=1:10224,rs6=1:10680,rs7=1:11256,rs8=1:11040,
          rs9=1:11784, rs10=1:12360,rs11=1:12480,
          rs12=1:12888,rs12=1:13416,rs14=1:13536,
```

```
          rs15 =1:14232 ,rs16 =1:14424 ,rs17 =1:14832 ,
          rs18 =1:15240 ,rs19 =1:15408 ,rs20 =1:16032 ,
          rs21 =1:16176 ,  rs22 =1:16560 ,rs23 =1:17136 ,
          rs24 =1:17400)
pred_on<-list (rs1 =8953:8777 ,rs2 =9164:9188 ,rs3 =9696:9720 ,
    rs4 =10129:10153 ,  rs5  =10225:10249 ,rs6 =10681:10705 ,
        rs7 =11257:11281 ,rs8 =11041:11064 ,  rs9 =11784:11808 ,
        rs10  =  12361:12385 ,  rs11  =  12480:12504 ,
        rs12 =12889:12913 ,rs13 =13417:13441 ,rs14 =13537:13561 ,
        rs15 =14233:14257 ,rs16 =14425:14449 ,rs17 =14832:14856 ,
        rs18 =15241:15265 ,rs19 =15409:15433 ,rs20 =16033:16057 ,
        rs21 =16176:16200 ,rs22 =16560:16584 ,rs23 =17136:17160 ,
        rs24 =17400:17424)
# mudel
# oma_grid  =  expand.grid(C  =  4,  sigma  =  4.088416  )
ctrl <- trainControl(method = "cv",index= fit_on, indexOut = pred_on,savePredictions =
    TRUE) # meetod ei oluline , kuna ise kindlask määratud valimid
mod_eps <- train(tarbimine~ ., data = treening_transf , method = "svmRadial", trControl =
     ctrl ,scaled = FALSE, epsilon = 0.02) # , tuneGrid = oma_grid
# salvestatud mudelid
# save(mod, file = "TVR2_mudel_valik.Rdata")
# save(mod_eps, file = "TVR2_mudel_uus_eps.Rdata")
# prognoos ja tagasi teisendamine
prognoos = predict(mod, test_transf)
tarb_kesk = mean(treening$tarbimine)
tarb_sd = sd(treening$tarbimine)
prognoos = prognoos*tarb_sd + tarb_kesk
# headus
mape = mean((abs(jaanuarveebruar$Tarbimine-prognoos))/jaanuarveebruar$Tarbimine)
mape = mean((abs(jaanuarveebruar$Tarbimine[169:length(prognoos)]-prognoos[169:length(
    prognoos)]))/jaanuarveebruar$Tarbimine[169:length(prognoos)])
rmse = sqrt((1/length(prognoos))*sum((jaanuarveebruar$Tarbimine-prognoos)**2))
```

C.3 Tugivektorregressiooni mudeli 3 kood

```
setwd("C:/Users/Lemz/Documents/Cliona_kraam/Kool/Magistritöö")
jaanuarveebruar = read.table("jaanuarveebruar1.csv", header = T,sep = ";", dec = ",")
attach(jaanuarveebruar)
treening = read.table("koosmoju_treening", header = T,sep = ",", dec = ".")
treening$X = NULL
test = read.table("koosmoju_test", header = T,sep = ",", dec = ".")
test$X = NULL
library(caret)
# skaleerimine
preProcvalues = preProcess(treening, method =  c("scale", "center"))
treening_transf = predict(preProcvalues, treening)
test_transf =  predict(preProcvalues, test)
test_transf$tarbimine = NULL
# mudeli tegemine
fit_on <-list (rs1 =1:8952 ,rs2 =1:9163 ,rs3 =1:9696 ,rs4 =1:10128 ,
          rs5 =1:10224 ,rs6 =1:10680 ,rs7 =1:11256 ,rs8 =1:11040 ,
          rs9 =1:11784 ,  rs10 =1:12360 ,rs11 =1:12480 ,
          rs12 =1:12888 ,rs12 =1:13416 ,rs14 =1:13536 ,
          rs15 =1:14232 ,rs16 =1:14424 ,rs17 =1:14832 ,
          rs18 =1:15240 ,rs19 =1:15408 ,rs20 =1:16032 ,
          rs21 =1:16176 ,  rs22 =1:16560 ,rs23 =1:17136 ,
          rs24 =1:17400)
pred_on<-list (rs1 =8953:8777 ,rs2 =9164:9188 ,rs3 =9696:9720 ,
    rs4 =10129:10153 ,  rs5  =10225:10249 ,rs6 =10681:10705 ,
        rs7 =11257:11281 ,rs8 =11041:11064 ,  rs9 =11784:11808 ,
        rs10  =  12361:12385 ,  rs11  =  12480:12504 ,
        rs12 =12889:12913 ,rs13 =13417:13441 ,rs14 =13537:13561 ,
        rs15 =14233:14257 ,rs16 =14425:14449 ,rs17 =14832:14856 ,
        rs18 =15241:15265 ,rs19 =15409:15433 ,rs20 =16033:16057 ,
        rs21 =16176:16200 ,rs22 =16560:16584 ,rs23 =17136:17160 ,
```

```
        rs24 =17400:17424)
# mudel
aeg_control <- trainControl(method = "cv", index = fit_on, indexOut = pred_on)
#oma_grid = expand.grid(C = 60, sigma = 0.3635339)
oma_grid = expand.grid(C = 3, sigma = 1.42)
SVR_test <- train(tarbimine ~ eelmine + tunnid*paev,
             data = treening_transf,
             method =  "svmRadial",
             trControl = aeg_control,
             scaled = FALSE, tuneGrid = oma_grid, epsilon = 0.0261) #  , tuneLength = 5,
# salvestatud mudelid "TVR3eps002.Rdata" ja "KoosmojuC60.Rdata"
# laadimine
mudel = load("KoosmojuC60.Rdata")
tugivek_koos$finalModel
prognoos = predict(SVR_test, test_transf)
std = sd(treening$tarbimine)
keskv = mean(treening$tarbimine)
prognoos= prognoos*std + keskv
mape = mean((abs(jaanuarveebruar$Tarbimine-prognoos))/jaanuarveebruar$Tarbimine)
mape = mean((abs(jaanuarveebruar$Tarbimine[169:length(prognoos)]-prognoos[169:length(
    prognoos)]))/jaanuarveebruar$Tarbimine[169:length(prognoos)])
rmse = sqrt((1/length(prognoos))*sum((jaanuarveebruar$Tarbimine-prognoos)**2))
```

D Mudelite parandamise kood

D.1 Uute suhete arvutamine, indikaatorite uuendamine

```
setwd("C:/Users/Lemz/Documents/Cliona kraam/Kool/Magistritöö")
# ANDMETE LUGEMINE
yl1.1 = read.table("prognoosiks2.csv", header = T, sep = ",", dec = ".")
yl1.1$X = NULL
attach(yl1.1)
# install.packages("lubridate")
# RIIGIPüHADE JA PUHKEPäEVADE TEGEMINE
paevad=(1:730)%%7
puhkepaevad=rep(0,730)
puhkepaevad[paevad>4]=1 #nädalavahetused
# pyhad=rep(0,730) teeme ainult pühade kohta indikaatori
library(lubridate) #et kuupäevale vastavat indeksit arvutada
puhkepaevad[yday(c("2013-01-01","2013-02-24","2013-03-29","2013-03-31","2013-05-01","
    2013-05-19",
"2013-06-23","2013-06-24","2013-08-20","2013-12-24","2013-12-25","2013-12-26"))]=1
puhkepaevad[365+yday(c("2014-01-01","2014-02-24","2014-04-18","2014-04-20","2014-05-01",
                       "2014-06-08","2014-06-23","2014-06-24","2014-08-20","2014-12-24",
                       "2014-12-25","2014-12-26"))]=1
# VIIMASE SAMA TüüPI PäEVAGA SUHETE MAATRIKS
ridu = 730
viimane_sama = matrix(NA, ridu, 24)
# algväärtustame osa
for ( i in 1:8){
  viimane_sama[i,]= yl1.1$tarbimine[(24*(i-1)+1):(24*i)]/tarbimine24[24*i]
}
# ülejäänud
for ( i in 9:ridu){
  j = i-1 # viimane sama tüüpi päev, kui on eelnev päev, siis jääb see
  while (puhkepaevad[i] != puhkepaevad[j]){ # ei tee vahet, kas on töö või puhke, peab
      olema sama tüüpi
    j = j-1 # kui on varasem päev, siis otsib üles
  }
  k = j-1 # eelviimane sama tüüpi päev
  while (puhkepaevad[j]!=puhkepaevad[k]){
    k = k-1
  }
```

```
  alpha = 0.35
  viimane_sama[i,]= alpha *yl1.1$tarbimine[(24*(j-1)+1):(24*j)]/tarbimine24[24*j] + (1-
      alpha)* viimane_sama[k,]
}
# R->L
yl1.1$indikaatorL = as.numeric(yl1.1$paev == "laupäev")
# modifikatsioon, anname laupäeva indikaatorile riigipühade väärtused juurde
pyhad=rep(0,730) #teeme ainult pühade kohta indikaatori
pyhad[yday(c("2013-01-01","2013-02-24","2013-03-29","2013-03-31","2013-05-01","
    2013-05-19",
"2013-06-23","2013-06-24","2013-08-20","2013-12-24","2013-12-25","2013-12-26"))]=1
pyhad[365+yday(c("2014-01-01","2014-02-24","2014-04-18","2014-04-20","2014-05-01",
                          "2014-06-08","2014-06-23","2014-06-24","2014-08-20","2014-12-24",
                          "2014-12-25","2014-12-26"))]=1
pyhad_pikk = rep(pyhad, each = 24) # tundide peale
sum(pyhad) # 24 päeva, 2 aastat
for (i in 1:length(pyhad_pikk)){
  if ((yl1.1$indikaatorL[i] == 0) & (pyhad_pikk[i] == 1) & (yl1.1$paev[i] != "pühapäev"
      )){
    yl1.1$indikaatorL[i] = 1
  }
}
# P->E
yl1.1$indikaatorE = as.numeric(yl1.1$paev == "esmaspäev")
# modifikatsioon, anname esmaspäeva indikaatorile riigipühade väärtused juurde
pyhad_pikk = rep(pyhad, each = 24) # tundide peale
sum(pyhad) # 24 päevad, 2 aastat
indikaatorE2 = rep(0, length(yl1.1$indikaatorE))
for (i in 1:(length(pyhad_pikk)-24)){
  if ((yl1.1$indikaatorE[i] == 0) & (yl1.1$indikaatorE[i+24] == 0) & (pyhad_pikk[i] ==
      1) & (yl1.1$paev[i] != "laupäev")){
    indikaatorE2[i+24] = 1
  }
}
yl1.1$indikaatorE = yl1.1$indikaatorE + indikaatorE2
tunnisuhe = as.vector(t(viimane_sama)) # ühte veergu tunnisuhted
# eelmise ööpäeva tarbimine
tarbimine24_uus = rep(NA,length(yl1.1$tunnid))
tarbimine24_uus[1:24] = tarbimine24[1:24] # esimene ööpäev jääb samaks
i = 25
for (i in 25:17520){
  tarbimine24_uus[i] = yl1.1$tarbimine24[i-24]
  i = i +1
}
korrutisL = yl1.1$indikaatorL*tunnisuhe
korrutisE = yl1.1$indikaatorE*tunnisuhe
korrutis2 = tarbimine24_uus*tunnisuhe
# teeme andmestiku, kust pealt mudel teha
andmestik = data.frame(kuupaevad, tarbimine, korrutis2, korrutisE, korrutisL)
# salvestame
#write.csv(andmestik, "suhted_riigipuhad_trenn.csv")
#write.csv(andmestik, "suhted_riigipuhad_trenn_lisa.csv")
#write.csv(andmestik, "suhted_riigipuhad_trenn_lisa_uus.csv")

######## JAANUARI-VEERBUARI JAOKS SAMA
jaanuarveebruar = read.table("jaanuarveebruar1.csv", header = T,sep = ";", dec = ",")
paevad=c(3,4,5,6,(7:59))%%7
puhkepaevad=rep(0,59)
puhkepaevad[paevad>4]=1 # nädalavahetused
puhkepaevad[yday(c("2015-01-01","2015-02-24"))]=1
pyhad = rep(0,59)
pyhad[yday(c("2015-01-01","2015-02-24"))]=1
kuupaevad = rep(seq(as.Date("2015-01-01"), as.Date("2015-02-28"), by="days"), each = 24)
paev = weekdays(as.Date(kuupaevad))
# R->L
indikaatorL = as.numeric(paev == "laupäev")
# modifikatsioon, anname laupäeva indikaatorile riigipühade väärtused juurde
```

```r
pyhad_pikk = rep(pyhad, each = 24) # tundide peale
sum(pyhad) # 2 päeva, 1 jaanuar, 24 veebruar
for (i in 1:length(pyhad_pikk)){
  if ((indikaatorL[i] == 0) & (pyhad_pikk[i] == 1) & (paev[i] != "pühapäev")){
    indikaatorL[i] = 1
  }
}
# P->E
indikaatorE = as.numeric(paev == "esmaspäev")
# modifikatsioon, anname esmaspäeva indikaatorile riigipühade väärtused juurde
pyhad_pikk = rep(pyhad, each = 24) # tundide peale
sum(pyhad) # 24 päevad, 2 aastat
indikaatorE2 = rep(0, length(indikaatorE))
for (i in 1:(length(pyhad_pikk)-24)){
  if ((indikaatorE[i] == 0) & (indikaatorE[i+24] == 0) & (pyhad_pikk[i] == 1) & (paev[i]
      != "laupäev")){
    indikaatorE2[i+24] = 1
  }
}
sum(indikaatorE2) # 2 sellist päeva
indikaatorE = indikaatorE + indikaatorE2
# käesoleva 24 h tarbimine
tarbimine24J = rep(aggregate(jaanuarveebruar$Tarbimine, by=list(0:(length(
    jaanuarveebruar$Tarbimine)-1)%/%24), sum)$x, each = 24)
# VIIMASE SAMA TüüPI PäEVA MAATRIKS
# võtame osa suhete väärtusi 2014 aasta lõpust
alpha = 0.35
dets27 = viimane_sama[726,]
dets28 = viimane_sama[727,]
dets29 = viimane_sama[728,]
dets30 = viimane_sama[729,]
dets31 = viimane_sama[730,]
ridu = 31+28
viimane_sama = matrix(NA, ridu, 24)
# algväärtustame osa
viimane_sama[1,] = alpha*dets27 + (1-alpha)*dets28
viimane_sama[2,] = alpha*dets30 + (1-alpha)*dets31
viimane_sama[3,] = alpha*dets28 + (1-alpha)*viimane_sama[1,]
viimane_sama[4,] = alpha*viimane_sama[1,] + (1-alpha)*viimane_sama[3,]
viimane_sama[5,] = alpha*dets31 + (1-alpha)*viimane_sama[2,]
# ülejäänud
for ( i in 6:ridu){
  j = i-1 # viimane sama tüüpi päev, kui on eelnev päev, siis jääb see
  while (puhkepaevad[i] != puhkepaevad[j]){ # ei tee vahet, kas on töö või puhke, peab
      olema sama tüüpi
    j = j-1 # kui on varasem päev, siis otsib üles
  }
  k = j-1 # eelviimane sama tüüpi päev
  while (puhkepaevad[j]!=puhkepaevad[k]){
    k = k-1
  }
  alpha = 0.35 # võtame kahe viimase sama tüüpi päeva kaalutud tunnisuhte
  viimane_sama[i,]= alpha *jaanuarveebruar$Tarbimine[(24*(j-1)+1):(24*j)]/tarbimine24J
      [24*j] +
    (1-alpha)* viimane_sama[k,] # jaanuarveebruar$Tarbimine[(24*(k-1)+1):(24*k)]/
        tarbimine24J[24*k]
  # üldine eksponenetsiaalne silumine
}
tundide_arv = (31+28)*24
tunnisuhe = as.vector(t(viimane_sama)) # ühte veergu tunnisuhted
# eelneva 24 h tarbimine
tarbimine24_uus = rep(NA,length(tarbimine24J))
tarbimine24_uus[1:24] = sum(yl1.1$Tarbimine[17497:17520])
i = 25
for (i in 25:tundide_arv){
  tarbimine24_uus[i] = tarbimine24J[i-24]
  i = i +1
```

```
}
korrutisL = indikaatorL*tunnisuhe
korrutisE = indikaatorE*tunnisuhe
korrutis2 = tarbimine24_uus*tunnisuhe
tarbimine = jaanuarveebruar$Tarbimine
uusdata = data.frame(kuupaevad, tarbimine, korrutis2, korrutisE, korrutisL)
# salvestame
# write.csv(uusdata, "suhetega_riigipuhad_test_lisa")
# write.csv(uusdata, "suhetega_riigipuhad_test_lisa_uus")

######## MäRTSI-APRILLI JAOKS SAMA
martsaprill = read.table("3kuud.csv", header = T,sep = ";", dec = ",")
# library(lubridate)
paevad=c(3,4,5,6,(7:120))%%7 # 2015
puhkepaevad=rep(0,120)
puhkepaevad[paevad>4]=1 # nädalavahetused
puhkepaevad[yday(c("2015-04-03","2015-04-05"))]=1
pyhad = rep(0,120)
pyhad[yday(c("2015-04-03","2015-04-05"))]=1
pyhad = pyhad[60:120]
puhkepaevad = puhkepaevad[60:120]
kuupaevad = rep(seq(as.Date("2015-03-01"), as.Date("2015-04-30"), by="days"), each = 24)
paev = weekdays(as.Date(kuupaevad))
# edasi analoogselt jaanuari-veebruariga
# algväärtused teised
veebr22 = viimane_sama[53,]
veebr26 = viimane_sama[57,]
veebr27 = viimane_sama[58,]
veebr28 = viimane_sama[59,]
ridu = 31+30
viimane_sama = matrix(NA, ridu, 24)
# algväärtustame osa
viimane_sama[1,] = alpha*veebr22 + (1-alpha)*veebr28
viimane_sama[2,] = alpha*veebr26+ (1-alpha)*veebr27
viimane_sama[3,] = alpha*veebr27 + (1-alpha)*viimane_sama[2,]
viimane_sama[4,] = alpha*viimane_sama[2,] + (1-alpha)*viimane_sama[3,]
viimane_sama[5,] = alpha*viimane_sama[3,] + (1-alpha)*viimane_sama[4,]
viimane_sama[6,] = alpha*viimane_sama[4,] + (1-alpha)*viimane_sama[5,]
viimane_sama[7,] = alpha*veebr28 + (1-alpha)*viimane_sama[1,]
# salvestame
# write.csv(uusdata1, "suhetega_riigipuhad_martsaprill")
# write.csv(uusdata1, "suhetega_riigipuhad_martsaprill_lisa")
# write.csv(uusdata1, "suhetega_riigipuhad_martsaprill_lisa_uus")
```

Viited

[1] Miloš Božic, Miloš Stojanovic, and Zoran Stajic. Short-Term Electric Load Forecasting Using Least Squares Support Vector Machines. *FACTA UNIVERSITATIS Series: Automatic Control and Robotics*, 9(1):141–150, 2010.

[2] Valdimir Cherkassky and Yunqian Ma. Practical Selection of svm Parameters and Noise Estimation for svm Regression. *Neural Networks*, 17(1):113–126, 2004.

[3] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[4] Elering. Eesti elektrisÜsteemi varustuskindluse aruanne. `http://elering.ee/public/Infokeskus/Aruanded/Elering_Varustuskindluse_aruanne_2011.pdf`. [10.05.2015].

[5] Ricardo Gutierrez-Osuna. Intelligent Sensor Systems: lecture 13. `http://research.cs.tamu.edu/prism/lectures/iss/iss_l13.pdf`. [05.03.2015].

[6] Hyndman and Athanasopoulos. Evaluating forecast accuracy. `https://www.otexts.org/fpp/2/5`, 2013. [20.03.2015].

[7] Raul Kangro. E-kursuse Äegridade Analüüs"materjalid. `http://dspace.utlib.ee/dspace/bitstream/handle/10062/27703/aegridede_analuus.pdf?sequence=1`, 2012. [15.04.2015].

[8] Alexandros Karatzoglou, David Meyer, and Kurt Hornik. Support Vector Machines in R. *Journal of Statistical Software*, 15(9), 2006.

[9] Kadir Kavaklioglu. Modeling and prediction of Turkey's electricity consumption using Support Vector Regression. *Applied Energy*, 88(1):368–375, 2011.

[10] Max Kuhn. Data splitting. `http://topepo.github.io/caret/splitting.html#time`. [01.04.2015].

[11] Max Kuhn. Building Predictive Models in R Using the caret Package. *Journal of Statistical Software*, 28(5), 2008.

[12] Vassili Mušnikov. On The Usage Of Support Vector Machines For Short-Term Price Movement Prediction In Intra-Day Trading. Master's thesis, University of Tartu, 2013.

[13] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels.* The MIT Press, Cambridge, Massachusetts, 2002.

[14] Romain Serres. Short-Term Load Forecasting in New York State: NYISO method and Support Vector Regression. `http://www.ee.columbia.edu/~lavaei/Projects/Romain_Serres.pdf`. [01.03.2015].

[15] Alex J. Smola and Bernhard Schölkopf. A Tutorial on Support Vector Regression. *Statistics and Computing 14*, 14:199–222, 2004.

[16] Jose Guajardo Sven F.Crone and Richard Weber. The Impact of Preprocessing on Support Vector Regression and N eural Networks in Time Series Prediction. In *Proceedings of the International Conference on Data Mining DMIN'06 (Las Vegas, USA) 2006, CSREA, pp.37-42*, pages 37–42, 2006.

[17] Shu xia Yang and Yi Wang. Applying Support Vector Machine Method to Forecast Electricity Consumption. In *2006 International Conference on Computational Intelligence and Security*, number 1, 2006.

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Cliona Georgia Dalberg,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose

"Eesti elektri tarbimise prognoos,"

mille juhendajad on Raul Kangro ja Joosep Lassmann.

1.1. reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
1.2. üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace′i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, 13.05.2015