TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

JAAKKO PIHLAJASALO

IMPROVEMENT OF SATELLITE ORBIT PREDICTION ACCURACY AND QUALITY WITH DEEP LEARNING AND SPECTRAL ANALYSIS

Master of Science thesis

Examiners: Ph.D. Helena Leppäkoski, Ph.D. Simo Ali-Löytty and Prof. Robert Piché
Examiners and topic approved
on 27th August 2017

ABSTRACT

JAAKKO PIHLAJASALO: Improvement of satellite orbit prediction accuracy and quality with deep learning and spectral analysis
Tampere University of Technology
Master of Science thesis, 47 pages, 0 Appendix pages
October 2017
Master's Degree Programme in Science and Engineering
Major: Mathematics
Examiner: Ph.D. Helena Leppäkoski, Ph.D. Simo Ali-Löytty and Prof. Robert Piché
Keywords: GNSS, GPS, BeiDou, satellite orbit predictions, deep learning, convolutional neural networks, RTN, spectral analysis, periodogram, quality monitoring system, health prediction

In this study, we consider methods of spectral analysis and deep learning to improve accuracy and quality of GNSS satellite orbit predictions. The quality of predictions decreases when satellite's orbit is fixed with maneuvers, which causes satellite orbit predictions to fail. In our previous research orbit accuracy has been improved using analytical and data-driven models.

The goal of this study is to improve quality of BeiDou satellite predictions with spectral analysis and to improve accuracy of GNSS satellite orbit predictions with deep learning. Both methods are used to improve the existing model for satellite orbit predictions. The improvement in quality is done by predicting unhealthy time periods of BeiDou's GEO and IGSO satellites. Periodogram was used to estimate a period of health parameter from broadcast. The improvement in accuracy was done with convolutional neural networks. The convolutional neural network was used to predict RTN errors of existing models orbit predictions. The errors were predited from RTN errors from the beginning of the orbit prediction. With these error predictions the predicted orbit can be corrected.

The main results of this study were that unhealthy periods can be predicted from the broadcast data and even a simple convolutional neural network can improve orbit prediction accuracy significantly. Health prediction algorithm could be created from the estimated periods to predict health successfully. Health prediction worked with BeiDou's GEO and IGSO satellites and predicted over 70% of unhealthy periods in testing. Two different methods for error correction with convolutional neural networks were created and tested with GPS satellites. The better method was also tested with BeiDou satellites of all orbit types. The method improves two week orbit prediction accuracy over 40% on average for all GPS and BeiDou satellites.

The best improvements in accuracy were achieved with GEO and IGSO satellites.

TIIVISTELMÄ

JAAKKO PIHLAJASALO: Satelliittien rataennustuksen tarkkuuden ja laadun parantaminen spektrianalyysin ja syväoppimisen menetelmillä
Tampereen teknillinen yliopisto
Diplomityö, 47 sivua, 0 Appendix pages
Lokakuu 2017
Teknis-luonnontieteellinen koulutusohjelma
Pääaine: Matematiikka
Tarkastajat: TkT Helena Leppäkoski, TkT Simo Ali-Löytty ja prof. Robert Piché
Avainsanat: GNSS, GPS, BeiDou, satelliittien rataennustus, syväoppiminen, konvoluutioneuroverkot, RTN, spektrianalyysi, periodogram, laadunvalvontajärjestelmä, terveysennustus

Tässä työssä tutkittiin spektrianalyysin ja syväoppimisen menetelmien käyttöä paikannussatelliittien rataennustuksien tarkkuuden ja laadun parantamiseen. Ennustuksien laatu vähenee kun satelliitin rataa korjataan manööverillä, joka aiheuttaa epäonnistuneita rataennustuksia. Rataennustuksen tarkkuutta on aikaisemin parannettu analyyttisillä ja dataan perustuvilla malleilla.

Tämän työn tavoitteena on parantaa BeiDou satelliittien rataennustuksien laatua spektrianalyysin avulla ja parantaapaikannussatelliittie rataennustuksien tarkkuutta syväoppimisen menetelmillä. Molempia menetelmiä käytetään jo olemassa olevan mallin parantamiseen. BeiDou satelliittien rataennustuksien laatua parannetaan ennustamalla ajanhetkiä, jolloin GEO ja IGSO satelliittien lähettämä data ei ole käyttökelpoista. Konvoluutioneuroverkkoja käytettiin tarkkuuden parantamiseen. Konvoluutioverkolla ennustettiin olemassa olevan mallin ennustuksen RTN -virheet rataennustuksen alun virheistä. Näillä virhe-ennustuksilla rataennustus voidaan korjata.

Tutkimuksen tuloksena oli, että rataennustuksen huonot ajanjaksot on mahdollista ennustaa lähetetystä datasta ja yksinkertainen konvoluutioverkko parantaa rataennustuksien tarkkuutta huomattavasti. Edellisen huonon ajanjakson avulla pystytään ennustamaan seuraava vastaava ajanjakso onnistuneesti. Terveysennustus toimii BeiDoun GEO ja IGSO ratatyypin satelliiteilla ja sillä onnistuttiin ennustamaan yli 70% huonoista ennustuksen ajanjaksoista testeissä. Kaksi erilaista menetelmää kehitettiin ja testattiin konvoluutioneuroverkkojen hyödyntämisestä GPS satelliiteille. Menetelmistä parempaa testattiin myös BeiDoun satelliiteilla. Menetelmä parantaa kahden viikon rataennustuksen tarkkuutta keskimäärin yli 40% kaikille GPS ja BeiDou satelliiteille. Suurimmat parannukset tarkuudessa saatiin BeiDoun GEO ja

IGSO satelliiteilla.

PREFACE

I started working in the Positioning Algorithms Group at the Laboratory of Automation and Hydraulic egineering in December 2016. Since then I have worked on topics of quality management and orbit prediction of GNSS satellites. The topic of this thesis arose during spring of 2016 from the project that our group and HERE have been collaborating. HERE has also funded this study.

I started working on this Master's thesis during spring of 2017 after a discussion with my supervisors. I started writing and testing process during the summer of 2017. Deep learning was completely new subject for me at the beginning of this work and thus I needed to familiarize myself with the topics. This was completely new territory for me, but I was enthusiastic to learn more about subjects of deep learning.

I would like thank my supervisors Helena Leppäkoski, Simo Ali-Löytty and Robert Piché for giving me ideas and encouragement to write this thesis. I am also grateful for the personnel that I work and have worked with in our research group for being an inspiration. I escpecially want to thank Sakari Rautalin for continuous support.

I also want to express my gratitude for all my friends outside work. For all my friends in the Guild of Science and Engineering Hiukkanen, thank you for conversations about everything and nothing and for your support during my studies in Tampere University of Technology. I also want to thank all of my friends outside of studies, who are and always have been there for me.

Tampere, 24th of October 2017

Jaakko Pihlajasalo

CONTENTS

LIST OF ABBREVIATIONS AND SYMBOLS

Abbreviations

BDS	BeiDou Navigation Satellite System
BE	Broadcast Ephemeris
BeiDou	BeiDou Weixing Daohang Xitong
CNN	Convolutional neural network
ECEF	Earth-Centered Earth-Fixed
ECI	Earth-centered inertial
EOP	Earth Orientation parameters
GEO	Geostationary orbit
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
ICRS	International Celestial Reference System
IGS	International GNSS Service
IGSO	Inclined Geosynchronous Orbit
iid	independent and identically distributed
JPL	Jet Propulsion Laboratory
LFM	Latent Force Model
MAP	maximum a posteriori
MEO	Medium-Earth Orbit
MGEX	Multi-GNSS Experiment
MSE	mean squared error
PE	Precise Ephemeris
PRN	pseudorandom noise
RNN	Recurrent neural network
ReLU	Rectified linear unit
RINEX	Receiver Independent Exchange Format
RGB	Red-Green-Blue color model
RMSE	root-mean-square error
RTN	Radial-Tangential-Normal
SGDM	Stochastic gradient descent with momentum
SISRE	Signal-In-Space-Range-Error
SRP	solar radiation pressure
UTC	Coordinated Universal Time
UT1	Universal Time

Notations

t, u, ...	scalars
$\mathbf{r}$, $\mathbf{v}$, ...	vectors
F, L, ...	matrices
$a \propto b$	a is proportional to b
$\|\cdot\|$	vector (2-)norm
$\mathbf{r}^T$	transpose of a vector or matrix
A^{-1}	inverse of a matrix A
$p(x)$	probability distribution function of a random variable x
$y\|x$	y with condition x
$\mathrm{N}(\mathbf{m}, \mathrm{P})$	multivariate normal distribution with mean $\mathbf{m}$ and covariance matrix P
$\det \mathrm{A}$	determinant of a matrix P
$\mathrm{A}_{[n,m]}$	element of matrix A in position (n, m).
$\hat{x}$	estimate of variable x
∇_θ	gradient in respect to variable θ
$\dot{\mathbf{r}}$	first derivative of $\mathbf{r}$ with respect to time
$\ddot{\mathbf{r}}$	second derivative of $\mathbf{r}$ with respect to time
$\sum_i$	sum of items over i
$\max(a, b)$	maximum value of values a and b
$\min(a, b)$	minimum value of values a and b
$\mathrm{relu}(x)$	rectified linear function of variable x
$\mathrm{maxpooling}_{(n\times m,S)}(\mathrm{A})$	max pooling of image A with filter size $n \times m$ and stride S
Δx	error in variable x
$\mathbf{a} \times \mathbf{b}$	cross product of vectors $\mathbf{a}$ and $\mathbf{b}$
$x \in X$	x is an element of set A
$\mathrm{A} \circ \mathrm{B}$	Hadamard product of matrices A and B
$\mathrm{A} * \mathrm{B}$	convolution of matrices A and B
$\mathrm{tr}(\mathrm{A})$	trace of a matrix A

Symbols

$\mathbf{1}_{n\times m}$	vector or matrix of ones with size $n \times m$
A	input image matrix of a convolutional layer
A_i	image's matrix of channel i
A_{Sat}	area of the Satellite
AU	average distance of the Earth from the Sun
a	parameter in posterior of multivariate normal linear model

a_i	output i of previous filter in a convolutional layer
$\mathbf{a}$	acceleration of the satellite
$\mathbf{a}_{\text{cb}}$	acceleration caused by the gravitation of a celestial body
$\mathbf{a}_{\text{Earth}}$	acceleration caused by the gravitation of the Earth
$\mathbf{a}_{\text{fcl}}$	output of fully connected layer
$\mathbf{a}_{\text{fcl,i}}$	output i of layer preceding fully connected layer
$\mathbf{a}_i$	acceleration of the satellite caused by the force i
$\mathbf{a}_{\text{input}}$	connected inputs of fully connected layer
$\mathbf{a}_{\text{SRP}}$	acceleration caused by solar radiation pressure
$\mathbf{a}_{\text{SRP,direct}}$	acceleration caused by solar radiation pressure to direction of the Sun
$\mathbf{a}_{\text{SRP,y-bias}}$	acceleration caused by solar radiation pressure to y-bias direction
α	momentum rate
α_1	parameter in solar radiation pressure model
α_2	parameter in solar radiation pressure model
B	filter matrix of a convolutional layer
B_i	filter's matrix of channel i
b	bias of neural networks layer
b_j	bias of j-th neuron
$\mathbf{b}_a$	amplitude parameter vector in observation model
$\mathbf{b}$	bias vector in neural networks
β	Beta angle
C	output matrix of a convolutional 2D layer
C_{nm}	coefficient of Earth Gravitational Model
$\text{CI}_{95\%}$	95% confidence interval
c	variable in posterior distribution of periodogram
c_1	amplitude parameter in a signal
c_2	amplitude parameter in a signal
$\mathbf{c}$	vector of signal amplitudes
γ	radiation pressure coefficient
d	number of color channels in image
Δ	sampling period
ΔN	normal error
ΔR	radial error
ΔT	tangential error
ΔT_i	tangential difference
$\mathbf{e}_{\text{H}}$	unit vector normal to the orbital plane of the satellite
$\mathbf{e}_{\text{N}}$	unit vector to normal direction
$\mathbf{e}_{\text{R}}$	unit vector to radial direction

$\mathbf{e}_{\text{Sat,Sun}}$	unit vector in the direction from the Satellite to the Sun
$\mathbf{e}_{\text{Sun}}$	unit vector in the direction from the Earth to the center of the Sun
$\mathbf{e}_{\text{T}}$	unit vector to tangential direction
$\mathbf{e}_{\text{y}}$	unit vector to the y-direction
η	learn rate
F_s	filter size of a neural network layer
$f(t_i)$	single stationary harmonic signal at time instant i
$\mathbf{f}(\mathbf{t})$	vector of single stationary harmonic signals for time instants $\mathbf{t}$
G	gravitational constant
$g(\cdot)$	activation function of neuron
I_n	Identity matrix of size n
λ	latitude of the satellite
N	normal position coordinate of the satellite
I	imaginary part of periodogram
I_s	input size of a neural network layer
$J(\theta)$	Mean squared error of weight parameters θ
M_{cb}	mass of a celestial body
M_{Earth}	mass of the Earth
m_{Sat}	mass of the satellite
$\mathbf{m}$	momentum vector of backpropagation
n	integer variable
n_j	response of j-th neuron
ν	shadow parameter
O_s	output size of a neural network layer
ω	angular frequency of the signal
P	zero padding of an input in convolutional neural network
P_{nm}	Legendre polynomials of degree n and order m
P_0	average solar radiation pressure at Earth's distance
ϕ	longitude of the satellite
R_{RTN}	transformation matrix from ECI to RTN
R_{RTN}	transformation matrix from ECI to ECEF
R	real part of periodogram
R_{E}	radius of the Earth
R	radial position coordinate of the satellite
r	distance of the satellite from Earth's center
$\mathbf{r}$	position of the satellite
$\mathbf{r}_0$	initial position of the satellite
$\mathbf{r}_{\text{cb}}$	position of a celestial body
$\mathbf{r}_{\text{Sat}}$	position of the satellite

$\mathbf{r}_{\text{Sun}}$	position of the Sun
S	stride of a filter in convolutional neural network
S_{nm}	coefficient of Earth Gravitational Model
s	variable in posterior distribution of periodogram
σ	standard deviation
σ^2	variance
$\mathbf{v}_0$	initial velocity of the satellite
T	tangential position coordinate of the satellite
T	period
t	time
t_i	time instant i
t_{toe}	time of ephemeris
$\mathbf{t}$	time vector of time instants t_i
$\boldsymbol{\theta}$	parameter vector
U_{E}	potential of the Earth's gravitation
W	weight matrix
w_{ji}	i-th weight of the j-th neuron
w_{R}	radial weight parameter of SISRE
$w_{\text{T,N}}$	tangential-normal weight parameter of SISRE
y_i	observation at a time instant i
$\mathbf{y}$	observation vector
$\mathbf{y}_i$	i-th response of a neural network
$\hat{\mathbf{y}}_i$	i-th prediction of a neural network
$\mathbf{y}_{ECI}$	coordinate vector in ECI-frame
$\hat{\mathbf{y}}_{ECI}$	estimated errors in ECI-frame
$\mathbf{y}_{\text{N}}$	vector of normal errors
$\mathbf{y}_{\text{R}}$	vector of normal errors
$\mathbf{y}_{RTN}$	coordinate vector in RTN-frame
$\hat{\mathbf{y}}_{RTN}$	estimated errors in RTN-frame
$\mathbf{y}_{\text{response}}$	response vector of neural network
$\mathbf{y}_{\text{T}}$	vector of normal errors
X	matrix of a signal sinusoids
x_i	i-th input to neuron

1. INTRODUCTION

When discussing positioning related subjects, GPS (Global Positioning System) is one of the first things that usually come into mind. Satellite positioning is used in many everyday applications with navigation of a vehicle, tracking luggage and locating yourself on a map to name a few. Because of satellite-based positioning's usefulness and importance, constellations of satellites, called GNSS (Global Navigation and Satellite System), are designed especially for positioning in mind. GPS is one of these constellations, but it isn't the only one.

To find the position of a user at least 4 satellites are needed. Position is based on three dimensions and the clock offset of the user. Satellites send a broadcast of their position and clock offset along with data that identifies the satellite and reliability of the data. The whole broadcast takes time and thus the positioning of the user takes longer. The broadcast is also not valid for a long time. The positioning of the user could thus take several minutes, which is not a wanted feature for positioning applications.

If the user has an Internet connection, one could download the ephemeris with position data and use that to locate the satellite faster. Thus with a stable internet connection the positioning is fast, but a stable connection is not granted in all situations. What we would like to do is extend the usability of one broadcast, such that one broadcast could be used days or even weeks after receiving the broadcast. This data can be used to predict the future positions and clock offsets of the satellite. We can use the broadcast as a initial state of the satellite and integrate this state to find out the positions of the satellite later.

The position of the satellite can also be found with laser ranging systems that give the position with high precision. We can compare our prediction results with these precise ephemerides. Our model for the motion of the satellite, is not perfect and thus position errors tend to happen. The errors can be explained for example by inaccuracies in our model, the inaccurate initial state from broadcast and as well as errors in equipment and atmosphere.

Our predictions rely on the data sent by the satellites and thus we need to be certain that data is reliable. If the satellites orbit is fixed with maneuvers during the orbit predictions, the errors of our prediction grow quite large quickly. By knowing time periods, during which the data is unreliable, the quality of our positioning increases.

By having better estimations of the satellite positions, the same initialization of the estimations could be used for longer time periods. Our model still shows imperfections which might not be able to be predicted with analytical methods. With state of the art performances and advancement with deep learning we attempt to use these methods to our predictions. We want to find out if deep learning could be used to improve either quality or accuracy of our method.

The goal of this Master's thesis is to improve the quality and accuracy of our method for satellite predictions using spectral analysis and deep learning. Some constellation's satellites make regular unannounced maneuvers to keep their satellites on their orbit. We are interested to find out if we could predict the times of these maneuvers to improve our quality of the predictions. The errors of our orbit prediction also show a periodic behavior and we want to find out if we could reduce these errors by predicting the errors with deep learning beforehand and fix the orbit prediction with these predicted error parameters. For both goals we use data which can be received directly from the satellite via regular broadcasts.

Our research group has previously studied different methods to improve orbit prediction accuracy and quality. Our force model of the satellites only use the forces that have highest impact on the movement of the satellite. We have tested an enhanced force model with more analytical forces modeled, but the model didn't significantly improve orbit prediction accuracy [24, 25].

Solar radiation pressure (SRP) is one of the largest forces affecting the satellites and thus the estimation of SRP parameters has been previously researched. In [8] an empirical two parameter SRP model is used to estimate the parameters of SRP. The angle between the satellite, Sun and Earth has also been noticed to affect the orbit prediction accuracy and thus a seasonal solar radiation pressure model has been studied [17].

Data-driven approaches to improve orbit prediction has also been tested in our research group. Latent forces can be estimated with either broadcast or precise data and these forces can then be applied to improve orbit predictions. Both methods have successfully improved satellite orbit prediction accuracy. [13, 27, 28]

BeiDou quality has been researched by comparing broadcast data to GPS satellites

broadcast data. In [14] differences between parameters during ephemeris update are studied between constellations. Also BeiDou orbit maneuvers and anomalies have successfully been detected from broadcast data previously. In [36] two broadcast ephemerides are used to compute the orbital coordinates of same reference epoch. Then first order differences of these positions are computed and compared to detection criteria to detect anomalies and maneuvers.

Different neural networks were studied to see which deep learning method would suit best for our research needs. One of the methods was to use neural networks as a black-box method to classify and spot patterns in our prediction accuracy. Another idea was to use neural networks to predict upcoming errors in our predictions with regression by using recurrent neural networks (RNNs) or convolutional neural networks (CNNs). Finally, convolutional neural networks were chosen to do further research on, since CNNs have shown success on other signal processing tasks [26, 34].

Deep learning methods have been previously used for positioning. Unlike in this study deep learning has been used for Wi-Fi positioning rather than GNSS positioning. In [37] Wi-Fi positioning is represented as a classification problem. Wireless signals were represented as vectors and a deep neural network was used to learn features from the signals and calculate the probabilities of discrete positions. CNNs have also been used for positioning tasks. In [35] CNNs were used to learn the sparse structure of massive MIMO channels for fingerprint-based positioning.

This study is structured as follows. First, in Chapter 2 we will introduce spectral analysis and deep learning techniques and methods, which are used in this study. In Chapter 3 our method for satellite orbit predictions is introduced, along with coordinate systems, models and sources of data. Health prediction for quality improvement with results is presented in Chapter 4. Chapter 5 focuses on implementing the deep learning methods to our orbit predictions and our results are presented. In Chapter 6 we give our conclusions and discuss future work related to his study.

2. THEORETICAL BACKGROUND

Here some mathematical tools are described, which are used to achieve our research goals. For spectral analysis used in health prediction, periodogram is introduced. Concept behind neural networks and especially convolutional neural network (CNN) is explained in greater detail.

2.1 Periodogram

Periodogram is a heuristic technique for detecting periodicity and estimating frequencies of a stationary signal. A Bayesian interpretation can be given to the periodogram. Periodogram is basically the squared modulus of complex Fast Fourier transforms of signals. [10]

Objective is to identify unknown period of a single stationary harmonic signal given noisy time series. Assume the observations y_i of the signal to be normally distributed with mean $f(t_i)$ and standard deviation σ.

Thus the model of the signal f is

$$f(t) = c_1 cos(\omega t) + c_2 sin(\omega t), \tag{2.1}$$

with an observation model

$$y_i|\omega, c_1, c_2, \sigma^2 \overset{\text{iid}}{\sim} \mathrm{N}(f(t_i), \sigma^2), \tag{2.2}$$

where c_1 and c_2 are amplitudes of the signal, ω is the angular frequency of the signal, y_i is an observation at a time instant t_i and σ^2 is the variance of the distribution. Notation $N(f(t_i), \sigma^2)$ is the normal distribution with mean $f(t_i)$ and variance σ^2. Sampling instants are equally-spaced in $t \in [-\frac{1}{2}T, \frac{1}{2}T]$ with sampling period $\Delta = T/(n-1)$, where n is number of observations and T is the period. [10, p. 13]

By collecting Equation (2.1) for all time instants t_i, we can form a matrix equation

$$\mathbf{f}(\mathbf{t}) = \begin{bmatrix} \cos(\omega t_1) & \sin(\omega t_1) \\ \cos(\omega t_2) & \sin(\omega t_2) \\ \vdots & \vdots \\ \cos(\omega t_n) & \sin(\omega t_n) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \mathrm{X}\mathbf{c}, \tag{2.3}$$

where X is a matrix of the signal sinusoids, $\mathbf{c}$ is a vector of signal amplitudes and

$$t_i = \frac{i-1}{n-1}T - \frac{T}{2} = \Delta(i-1) - \frac{T}{2}. \tag{2.4}$$

Now assuming conditionally independent samples, the observation model is

$$\mathbf{y}|\omega, \mathbf{c}, \sigma^2 \sim \mathrm{N}(\mathrm{X}\mathbf{c}, \sigma^2 \mathrm{I_n}), \tag{2.5}$$

where $\mathbf{y}$ is a vector of observations and I_n is the identity matrix of size n. [23]

As shown in [23], the posterior distribution $p(\omega|\mathbf{y})$ of our model is

$$p(\omega|\mathbf{y}) \propto \frac{p(\omega)}{\sqrt{\det \mathrm{X}^T\mathrm{X}}} \left(\|\mathbf{y}\|^2 - \mathbf{y}^T\mathrm{X}(\mathrm{X}^T\mathrm{X})^{-1}\mathrm{X}^T\mathbf{y}\right)^{1-n/2}. \tag{2.6}$$

We need $\mathrm{X}^T\mathbf{y}$ and $\mathrm{X}^T\mathrm{X}$, which is a symmetric 2×2 matrix. Starting with matrix

$$\mathrm{X}^T\mathrm{X} = \begin{bmatrix} \sum_{i=1}^n \cos^2(\omega t_i) & \sum_{i=1}^n \cos(\omega t_i)\sin(\omega t_i) \\ \sum_{i=1}^n \cos(\omega t_i)\sin(\omega t_i) & \sum_{i=1}^n \sin^2(\omega t_i) \end{bmatrix}. \tag{2.7}$$

We shall solve the matrix component-wise. Now using power-reduction formulae on the diagonal components we get

$$(\mathrm{X}^T\mathrm{X})_{[1,1]} = \sum_{i=1}^n \cos^2(\omega t_i) = \sum_{i=1}^n \frac{1+\cos(2\omega t_i)}{2} = \frac{n}{2} - \sum_{i=1}^n \frac{\cos(2\omega t_i)}{2} \tag{2.8}$$

and

$$(\mathrm{X}^T\mathrm{X})_{[2,2]} = \sum_{i=1}^n \sin^2(\omega t_i) = \sum_{i=1}^n \frac{1+\cos(2\omega t_i)}{2} = \frac{n}{2} + \sum_{i=1}^n \frac{\cos(2\omega t_i)}{2}, \tag{2.9}$$

and using double-angle of sine we get

$$(\mathrm{X}^T\mathrm{X})_{[1,2]} = (\mathrm{X}^T\mathrm{X})_{[2,1]} = \sum_{i=1}^{n} \cos(\omega t_i)\sin(\omega t_i) = \frac{1}{2}\sum_{i=1}^{n} \sin(2\omega t_i) \tag{2.10}$$

We need to solve the sums from the previous equations. Using Equation (2.4) and arithmetic progression of angles [16] we get

$$\begin{aligned}\sum_{i=1}^{n} \cos(2\omega t_i) &= \sum_{i=1}^{n} \cos(2\omega(i-1)\Delta - \omega T) \\ &= \frac{\sin(n\omega\Delta)}{\sin(\omega\Delta)} \cos(-\omega T + (n-1)\omega\Delta) \\ &= \frac{\sin(n\omega\Delta)}{\sin(\omega\Delta)} \cos(-\omega T + \omega T) \\ &= \frac{\sin(n\omega\Delta)}{\sin(\omega\Delta)}\end{aligned}$$

and

$$\begin{aligned}\sum_{i=1}^{n} \sin(2\omega t_i) &= \sum_{i=1}^{n} \sin(2\omega(i-1)\Delta - \omega T) \\ &= \frac{\sin(n\omega\Delta)}{\sin(\omega\Delta)} \sin(-\omega T + (n-1)\omega\Delta) \\ &= \frac{\sin(n\omega\Delta)}{\sin(\omega\Delta)} \sin(-\omega T + \omega T) \\ &= 0.\end{aligned}$$

Matrix $\mathrm{X}^T\mathrm{X}$ can now be written as

$$\mathrm{X}^T\mathrm{X} = \begin{bmatrix} \frac{n}{2} + \frac{\sin(n\omega\Delta)}{2\sin(\omega\Delta)} & 0 \\ 0 & \frac{n}{2} - \frac{\sin(n\omega\Delta)}{2\sin(\omega\Delta)} \end{bmatrix} = \begin{bmatrix} c & 0 \\ 0 & s \end{bmatrix}, \tag{2.11}$$

with

$$c = \frac{n}{2} + \frac{\sin(n\omega\Delta)}{2\sin(\omega\Delta)}, \quad s = \frac{n}{2} - \frac{\sin(n\omega\Delta)}{2\sin(\omega\Delta)}. \tag{2.12}$$

Since $\sin(n\omega\Delta)/\sin(\omega\Delta) \in [-1, 1]$ and the number of observations n can be assumed large, Equation (2.12) simplifies to

$$c \approx \frac{n}{2}, \quad s \approx \frac{n}{2}. \tag{2.13}$$

Now we shall compute the vector $\mathrm{X}^T\mathbf{y}$, which is a simple matrix multiplication.

$$\mathrm{X}^T\mathbf{y} = \sum_{i=1}^{n} \begin{bmatrix} y_i \cos(\omega t_i) \\ y_i \sin(\omega t_i) \end{bmatrix} = \begin{bmatrix} R \\ I \end{bmatrix}, \tag{2.14}$$

with

$$R = \sum_{i=1}^{n} y_i \cos(\omega t_i), \quad I = \sum_{i=1}^{n} y_i \sin(\omega t_i). \tag{2.15}$$

Now we can solve the matrix equation in Equation (2.6)

$$\mathbf{y}^T\mathrm{X}(\mathrm{X}^T\mathrm{X})^{-1}\mathrm{X}^T\mathbf{y} = \begin{bmatrix} R & I \end{bmatrix} \begin{bmatrix} 1/c & 0 \\ 0 & 1/s \end{bmatrix} \begin{bmatrix} R \\ I \end{bmatrix} = \frac{R^2}{c} + \frac{I^2}{s}.$$

Our posterior in (2.6) simplifies to

$$p(\omega|\mathbf{y}) \propto \frac{p(\omega)}{\sqrt{cs}} \left(\|\mathbf{y}\|^2 - \frac{R^2}{c} - \frac{I^2}{s} \right)^{1-n/2}. \tag{2.16}$$

By assuming flat prior $p(\omega) \propto 1$ and using the approximation in (2.13) with large n our posterior is

$$p(\omega|\mathbf{y}) \propto \frac{2}{n} \left(\|\mathbf{y}\|^2 - \frac{2R^2}{n} - \frac{2I^2}{n} \right)^{1-n/2}. \tag{2.17}$$

The posterior is normalized to have an area of one. Now the estimate is chosen to be the maximum a posteriori (MAP) on the current interval.

$$\hat{\omega} = \arg\max_{\omega} p(\omega|\mathbf{y}). \tag{2.18}$$

If the posterior distribution has a single sharp peak we can use Laplacian method in [23, p. 42] to approximate the distribution to be normally distributed. With this approximation we can calculate 95% confidence intervals for the estimate with

$$\mathrm{CI}_{95\%} = \hat{\omega} \pm 1.96\hat{\sigma}, \tag{2.19}$$

where $\mathrm{CI}_{95\%}$ is the 95% confidence interval and $\hat{\sigma}$ is the standard deviation of normal distribution with the same maximum value as our posterior distribution has.

2.2 Convolutional neural networks

Convolutional neural networks (CNN) are made up of neurons with learnable weights and biases. Neuron's response can be defined as

$$n_j = \sum_{i=1} g(w_{ji}x_i + b_j), \tag{2.20}$$

where n_j is the response of j-th neuron, g is the activation function of the neuron, x_i is the i-th input to the neuron, w_{ji} is the i-th weight of the neuron and b_j is is the bias of the neuron. The activation function can be a non-linaer function such as ReLU or in the case of CNNs a convolution. The response of the neuron is then fed as input to next neurons.

In CNNs neurons receive input, performs dot products and adds a non-linearity. Lastly the neurons fully connect to a vector of regression values. The networks goal is to optimize the weights of the neurons to give desired values as the output of the network.

CNNs basically follows the regular patterns of neural networks. Unlike regular neural networks, CNNs assume the inputs are images, which can reduce the number of parameters dramatically compared to non-image data [15]. In regular networks patterns in data need a parameter for each pixel in an image when CNNs try to find features or multiple pixels in data thus reducing the number of parameters. Convolutional neural networks can be called as feature identifiers. The filters contain some feature we want to identify from the images and we look through the image with that filter.

Architecture of an CNN consists of different layers and hidden layers. In hidden layers neurons transform the input and connect it to other layers. The neurons in hidden layers are connected to all neurons of the previous layer. Other layers perform some function to the input or reduce parameters by different means. Lastly the inputs are connected desired values of class or regression depending on the task. If the netowrk is suited for classification these desired values are probabilities of classes and in regression the values are the regression values representing the real data. [15]

2.2.1 Network layers

The network and its individual layers can be described as a black box or a function which gives an output for some input. The input images size I_s is $width \times height \times$

depth with depth being the number of color channels. For example depth of one is a grayscale image and depth of three is a red-green-blue (RGB) image. When using a filter we can calculate the width or height, named as output-size O_s further on, with

$$O_s = \frac{I_s - F_s + 2P}{S} + 1, \tag{2.21}$$

where I_s is width or height, F_s is corresponding filter size, P is padding and S is stride. Padding an image means adding zeros to the border of the image. Stride means movement step size of the filter across the image. When using stride, one must be careful to choose a number which gives an integer for the whole image to be covered. [21]

All convolutional neural networks have image input layer as the first layer. The layer takes the images as inputs and then zero-center normalizes the images. Zero-center normalization means that first the mean of the data is subtracted from the data and then the data is normalized with the standard deviation of the data. This process is done individually for each dimension of the data.

2-D convolution is computed for all channels of the image in these layers. Size of convolution filter is chosen to be equal size or smaller than size of the image. The convolution filter B is slid across the image A according to the chosen padding and stride. The main mathematical operation in convolution is the sum of Hadamard product elements divided by number of the elements in the matrix. Hadamard product is also known as Schur product. For matrices A and B of same dimension Hadamard product is given elementwise by

$$(\mathrm{A} \circ \mathrm{B})_{i,j} = (\mathrm{A})_{i,j}(\mathrm{B})_{i,j}, \tag{2.22}$$

where i and j span the size of the matrices. [9]

Sum of elements in the Hadamard product matrix can be computed as

$$\mathrm{tr}(\mathrm{AB}^T) = \mathbf{1}_{1\times m}(\mathrm{A} \circ \mathrm{B}^T)\mathbf{1}_{n\times 1}, \tag{2.23}$$

where m and n are the dimension sizes of matrices and $\mathbf{1}_{1\times m}$ and $\mathbf{1}_{n\times 1}$ are vectors of ones. [9]

Thus 2-D convolution for image A and filter B of same size, can be defined as

$$C = \frac{1}{dn}\sum_{i=1}^{d} \mathrm{tr}(\mathrm{A}_i\mathrm{B}_i^T) + b, \tag{2.24}$$

where C is the convolution matrix, n is the number of elements in one channel of the image, d represents the number of color channels and b is bias of the filter. A_i and B_i are the matrices of color channel i. When dealing with matrices of different sizes, the filter is slid across matrix A and the results form a matrix C with elements according to the sliding of the filter. [29]

$$\begin{bmatrix} -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & 0 & -1 & 1 & 0 & 1 \\ -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & 0 & 0 & 1 & 0 & 1 \\ -1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & -1 & -1 \\ 0 & -1 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 1 & -1 & -1 & 1 & 1 \\ -1 & 1 & -1 & 1 & -1 & 1 \\ -1 & 1 & 1 & -1 & -1 & 1 \\ -1 & 1 & -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & -1 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & -1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} -2 & -7 & 4 & -1 \\ -2 & 0 & 1 & 4 \\ 2 & 0 & 9 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 1 & 1 & 1 & 1 \\ -1 & 0 & 0 & 0 & 0 & 1 \\ -1 & -1 & -1 & -1 & 0 & 1 \\ -1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 0 \\ -1 & -1 & -1 \\ 0 & 0 & 0 \end{bmatrix}$$

Figure 2.1 *An example of convolutional layers function on an image with depth of 3. For example the bottom-right value of the output matrix can be calculated by fitting the filter on the image in the same position and calculating the values with Equation (2.24).*

In Figure 2.1 is an example of a convolution of an image. The matrices on the left present the image to be convolved and the matrices on the center present the convolution filter. The input image's size $I_s = 5 \times 6$ and the filter size $F_s = 3 \times 3$. The convolution has a stride $S = 1$, padding $P = 0$ and a bias $b = -0.05$. The filter fits in the image on twelve different positions and thus the output size $O_s = 3 \times 4$. Hadamard product is then calculated for all positions. Positive values in the output present recognized features from the filter. Negative values reveal parts of the image that the filter didn't fit on.

$$\frac{1}{3 \cdot 9} \begin{bmatrix} -2 & -7 & 4 & -1 \\ -2 & 0 & 1 & 4 \\ 2 & 0 & 9 & 1 \end{bmatrix} - 0.05 = \begin{bmatrix} -0.12 & -0.31 & 0.1 & -0.09 \\ -0.12 & -0.05 & -0.01 & 0.1 \\ 0.02 & -0.05 & 0.28 & -0.01 \end{bmatrix} \tag{2.25}$$

In (2.25) the convolution example presented in Figure 2.1 is continued. After computing the Hadamard products in all positions the output is divided by number of elements in the filter which is $3 \cdot 9 = 27$ and the bias $b = -0.05$ is added.

Since convolution is a linear operation and response variables of the network might vary non-linearly, a non-linearity needs to be added to the network. Without added non-linearity the network wouldn't be able to model non-linear data. Rectified linear unit (ReLU) is a non-linear activation function which is defined as

$$f(x) = \max(0, x), \tag{2.26}$$

where x is the element of the image to be non-linearized.[21]

$$\text{relu}\left(\begin{bmatrix} -0.12 & -0.31 & 0.1 & -0.09 \\ -0.12 & -0.05 & -0.01 & 0.1 \\ 0.02 & -0.05 & 0.28 & -0.01 \end{bmatrix}\right) = \begin{bmatrix} 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \\ 0.02 & 0 & 0.28 & 0 \end{bmatrix} \tag{2.27}$$

In (2.27) is presented an example of a rectified linear unit on the output matrix from the convolution example in 2.25. As we can see all negative values are changed to zero.

When processing large amounts of data down-sampling can be used to reduce data and reduce the computational cost of the process. One such method is pooling. Pooling is done by applying a desired filter to regions of the images. These filters down-sample the regions by sampling each region with desired effect and placing the sampled value to a new image. The filters are moved over the images in same fashion as convolution layers, either with overlapping or non-overlapping regions.

Commonly used pooling is max pooling, which takes maximum value of the regions it is ran over. It's quite common for being computationally cheap. It is also discussed that no pooling should be done at all [32]. Also pooling can cause loss of information, which is disadvantageous for regression with output size being larger than the input. [15, 21]

$$\text{maxpooling}_{(1\times 3,1)}\left(\begin{bmatrix} 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \\ 0.02 & 0 & 0.28 & 0 \end{bmatrix}\right) = \begin{bmatrix} 0.02 & 0 & 0.28 & 0.1 \end{bmatrix} \tag{2.28}$$

In (2.28) is presented an example of max pooling. Filter is chosen to be of size 1×3 with stride of one. Thus maximum of each column is chosen from the matrix to form the output.

In fully connected layer all the neurons of the network are connected. Outputs of all the previous layers are connected and used to form the final output of the network. Weights and biases are used similarly to convolution layer to combine the final output.

The outputs of the previous layer are put together and then the output of the fully connected layer is computed as a matrix multiplication similarly to convolution layer in Equation (2.24). The output $\mathbf{a}_{\text{fcl}}$ of fully connected layer is

$$\mathbf{a}_{\text{fcl}} = \text{W}\begin{bmatrix} \mathbf{a}_{\text{fcl},1} \\ \mathbf{a}_{\text{fcl},2} \\ \vdots \\ \mathbf{a}_{\text{fcl,n}} \end{bmatrix} + \mathbf{b} = \text{W}\mathbf{a}_{\text{input}} + \mathbf{b}, \tag{2.29}$$

where W is the weight matrix of the layer, $\mathbf{a}_i$ is previous layers output i, $\mathbf{b}$ is the bias vector of the layer and $\mathbf{a}_{\text{input}}$ is the connected outputs of the previous layer.

2.2.2 Training and testing

Training of the weights and biases in the network are computed with backward propagation of errors, or simply backpropagation. Weights and biases are initially chosen at random if no previous information is available about the network. Now images are propagated through the network layers and the resulting output vector is compared to response values with cost function. Since we are using the network for regression loss function is chosen to be mean squared error (MSE). MSE is defined as

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}(\hat{\mathbf{y}}_i - \mathbf{y}_i)^2, \tag{2.30}$$

where θ are all weights of the network, n is the number of predictions, $\hat{\mathbf{y}}_i$ is the i-th prediction and $\mathbf{y}_i$ is i-th response. Now that we have the errors we can use them

to improve our weights. We will use stochastic gradient descent with momentum (SGDM) to iterate our weights. SGDM tries to minimize the error using gradient descent, which means taking steps towards the gradient minimum. This is where we use the networks learn rate η and momentum rate α. Weight update is given by

$$\mathbf{m} = \eta\mathbf{m} + \alpha\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \tag{2.31}$$

and

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \mathbf{m}, \tag{2.32}$$

where $\mathbf{m}$ is the momentum vector and $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ is the gradient of the loss function. This step is continued until some ending criteria is met. Ending criteria can be time, batch accuracy or number of iterations to name a few.

After training, the network should be tested with a set of images that have not been used in training. This is to ensure that the network doesn't overfit to training images. The network is used to predict the responses which are then compared to real responses to compute the accuracy. For regression accuracy we choose root-mean-square error, which is defined as the square root of mean square error (2.30).

3. SATELLITE ORBIT PREDICTIONS

The orbit and motion of a satellite can be interpreted as an initial value problem. The initial position and velocity of the satellite can be acquired from data the satellite sends and the motion of the satellite is a dynamic system. Using this information we can present a force model for the motion of the satellite and form the dynamic model function.

The equation of motion of the satellite can be presented as the sum of the accelerations caused by different forces affecting the satellite with initial conditions.

$$\ddot{\mathbf{r}}(t) = \mathbf{a}_{\text{Earth}} + \mathbf{a}_{\text{Sun}} + \mathbf{a}_{\text{Moon}} + \mathbf{a}_{\text{SRP}}, \quad \mathbf{r}(t_0) = \mathbf{r}_0, \quad \dot{\mathbf{r}}(t_0) = \mathbf{v}_0, \tag{3.1}$$

where $\mathbf{a}_{\text{Earth}}$, $\mathbf{a}_{\text{Sun}}$, $\mathbf{a}_{\text{Moon}}$ and $\mathbf{a}_{\text{SRP}}$ are accelerations caused by the largest forces affecting the satellite and $\mathbf{r}_0$ and $\mathbf{v}_0$ are respectively position and velocity at the initial time t_0. Initial conditions are acquired from the initial state algorithm, which will be discussed later in this chapter. Using dynamic models for different forces we can predict position and velocity of a satellite at a later time t.

Largest forces and thus accelerations acting on the satellite are gravitation of the Earth, the Sun and the Moon as well as solar radiation pressure (SRP). These forces are only dependent on time and the position of the satellite.

We shall take a look at coordinate systems required for orbit prediction and different constellations studied in this research. Models for four largest forces affecting the satellite will be briefly presented. Also data sources for satellite positions and velocities as well as different parameters used in our predictions, such earth orientation parameters, will be presented.

3.1 Coordinate systems

When observing GNSS satellites and their orbits many coordinate systems are used. For example Earth's gravity is easier to model in earth-fixed coordinate system but motion of another body can be complex in that system. Thus many coordinate

systems and the transformation methods between the systems are needed. These systems are used when calculations in that system is better for simplicity and accuracy. We will not delve to deep in the coordinate systems, but will introduce the required systems used in predictions and analysis.

The user of the satellites is on Earth, so naturally a coordinate system that is tied to the surface of the Earth is of interest. Since the Earth is rotating, the coordinate system is also in rotation. One such coordinate system is called *Earth-Centered Earth-Fixed* (ECEF) coordinate system. Since ECEF is fixed to the surface of the Earth, a fixed point on the surface has a fixed set of coordinates.

To simplify the orbits of other celestial bodies such as the Sun and the Moon we need another coordinate system. Inertial coordinate systems are not in rotation, which is good for our purposes, since we fix our inertial coordinate system to the start of the prediction. We will also need an inertial coordinate system in which the orbits of mentioned celestial bodies are close to simple ellipses. Such system that we use is *International Celestial Reference System* (ICRS). The force model depends on Newton's laws of motion, which apply in inertial coordinate systems. [18].

A coordinate system that is mainly used in our study for error analysis is *Radial-Transverse-Normal* (RTN) coordinate system. The system is defined by the position and velocity of the satellite. In RTN coordinate system radial direction is defined as the direction from center of the earth to the satellite, normal direction is the direction of the cross product of the position and velocity of the satellite and transverse direction completes the right handed coordinate system.

3.2 Data used for predictions

Data which is used in our satellite orbit predictions is available directly from satellites and from the Internet. Main sources of data are broadcast and precise ephemerides (BE and PE) which contain satellite positions. We also use Earth Orientation Parameters (EOP) and Development Ephemerides (DE), which are available from the internet.

Broadcast ephemeris is the data package broadcast by the satellite, which contains data used to compute the satellite positions and velocities [5, 11]. It also contains date, clock and health parameters. Different satellite constellations send these ephemerides at different intervals. GPS updates the broadcast every two hours and BeiDou every hour. To simulate satellite predictions we use data previously broadcasted, which is available in RINEX-format (Receiver Independent Exchange) from

the Internet for free [4, 6]. The position data from BE has an accuracy of roughly a meter [3]. The velocity computed from BE is too inaccurate to be used as an initial state for our predictions [7]. Thus in our predictions we use filtering techniques such as Extended Kalman filter to make our predictions with our model more accurate. Initial state estimation is presented more detailed in [7] and [27].

Precise ephemerides position data is centimeter-level accurate position which is more accurate than BE, but is available hours or even days after the broadcast [3]. PE is more accurate than BE, which is based on predictions of the orbit while PE is based on post-processing of orbit observations. PE position data is thus better for comparison against our predictions. Precise data has satellite coordinates in ECEF, but satellite velocities are not provided. Precise data is available in SP3-format with 15 minute sampling-interval [31]. Data is provided by IGS and MGEX for free [4].

The Earth's orientation is not constant and thus we need parameters to describe these irregularities. Earth Orientation Parameters describe these irregularities as a function of time in ICRS coordinate system. Earth orientation data is freely available from International Earth Rotation and Reference Systems Service (IERS) [2].

Universal Time (UT1) is defined to match the Earth's orientation and Coordinated Universal Time (UTC) is defined by a collection of atomic clocks. The difference between the two is called dUT1. UT1 is not constant, since the Earth's rotation is irregular and thus leap seconds are added to UTC to keep the difference less than 0.9 seconds. The dUT1 is also available from IERS. [2]

Development Ephemerides contains position data of solar bodies in the form of Chebyshev polynomials. These positions are needed to model the acceleration caused by Sun and Moon. DE, which contains past and future position of Sun, Moon and planets, is available from Jet Propulsion Laboratory (JPL). [12]

3.3 Analytical forces

The largest forces affecting the satellite can be modeled as functions of time and position. In this study the equations used to model the forces are briefly present. The derivations of the equations will not be presented here, but they are available in [30] and in [19].

The gravitation of the Sun and the Moon can be modeled as point masses, since the distances from the satellite to these celestial bodies are reasonably far away. Thus

the acceleration $\mathbf{a}_{\text{cb}}$ caused by these forces relative to motion of the Earth is

$$\mathbf{a}_{\text{cb}} = GM_{\text{cb}} \left(\frac{\mathbf{r}_{\text{cb}} - \mathbf{r}_{\text{Sat}}}{\|\mathbf{r}_{\text{cb}} - \mathbf{r}_{\text{Sat}}\|^3} - \frac{\mathbf{r}_{\text{cb}}}{\|\mathbf{r}_{\text{cb}}\|^3} \right), \tag{3.2}$$

where G is the gravitational constant, M_{cb} is the mass of the celestial body and $\mathbf{r}_{\text{cb}}$ and $\mathbf{r}_{\text{Sat}}$ are the positions of celestial body and the satellite.

Since the satellite is relatively close to the Earth whose mass distribution is irregular, a simple model is not enough for the gravitation of the Earth. The potential of the Earth's gravitational field can be modeled with Legendre polynomials. The potential U_{E} is

$$U_{\text{E}} = \frac{GM_{\text{E}}}{r} \sum_{n=0}^{\infty} \sum_{m=0}^{n} \left[\left(\frac{R_{\text{E}}}{r} \right)^n P_{nm} \sin\phi \Big(C_{nm} \cos(m\lambda) + S_n m \sin(m\lambda) \Big) \right], \tag{3.3}$$

where r is distance of the satellite from Earth's center, M_{E} and R_{E} are Earth's mass and radius and λ and ϕ are longitude and latitude of the satellite. P_{nm} are Legendre polynomials of degree n and order m. Coefficients C_{nm} and S_{nm} can be acquired from EGM2008 [22].

Now the acceleration affecting the satellite is

$$\mathbf{a}_{\text{Earth}} = \mathrm{R}_{\text{ECEF}}^{-1} \nabla U_{\text{E}}, \tag{3.4}$$

where R_{ECEF} is the transformation matrix from inertial coordinate system to ECEF.

Finally, the solar radiation pressure is modeled in two directions. First direction is direct acceleration from the Sun and the other points along the satellite's solar panel axis.

The direct acceleration $\mathbf{a}_{\text{SRP,direct}}$ caused by SRP is

$$\mathbf{a}_{\text{SRP,direct}} = -\alpha_1 P_0 (1+\gamma) \frac{A_{\text{Sat}}}{m_{\text{Sat}}} \frac{AU^2}{\|\mathbf{r}_{\text{Sun}} - \mathbf{r}_{\text{Sat}}\|^2} \mathbf{e}_{\text{Sat,Sun}}, \tag{3.5}$$

where α_1 is a cofficient to be estimated for each satellite, P_0 is the average solar radiation pressure at Earth's distance, γ is the radiation pressure coefficient, A_{Sat} and m_{Sat} are the area and mass of the satellite and AU is the average distance between the Earth and the Sun [19]. Vectors $\mathbf{r}_{\text{Sat}}$ and $\mathbf{r}_{\text{Sun}}$ are the positions of the satellite and the Sun and $\mathbf{e}_{\text{Sat,Sun}}$ is the unit vector from the satellite to the Sun.

Some of the solar radiation reflects off from the satellite and causes a force to the

direction of the satellites solar panel axis. This direction and force are called y-direction and *y-bias*, the acceleration of which is computed as

$$\mathbf{a}_{\text{SRP,y-bias}} = 10^{-9}\alpha_2 \frac{\mathbf{r}_{\text{Sat}} \times (\mathbf{r}_{\text{Sun}} - \mathbf{r}_{\text{Sat}})}{\|\mathbf{r}_{\text{Sat}} \times (\mathbf{r}_{\text{Sun}} - \mathbf{r}_{\text{Sat}})\|} = 10^{-9}\alpha_2 \mathbf{e}_{\text{y}}, \tag{3.6}$$

where α_2 is a parameter which is also estimated for each satellite and $\mathbf{e}_{\text{y}}$ is the unit vector to y-direction.

Thus total acceleration is

$$\mathbf{a}_{\text{SRP}} = \nu(\mathbf{a}_{\text{SRP,direct}} + \mathbf{a}_{\text{SRP,y-bias}}), \tag{3.7}$$

where $\nu \in [0, 1]$ is a parameter which models the shadows of celestial objects.

The SRP parameters are estimated for a certain time period from precise ephemeris data. The parameters need to be updated either annually or biannually.

3.4 Constellations

Here satellite constellations considered in this study are shortly introduced. Main focus will be on the Global Positioning System (GPS) and BeiDou Navigation Satellite System (BDS or BeiDou). Properties such as orbit type and orbital period of satellites in these systems will be discussed.

Both systems have multiple satellites with a unique pseudorandom noise (PRN) code, which is transmitted as a part of the navigation message. PRN is used to uniquely identify satellites of the same constellation in this work.

3.4.1 GPS

Global Positioning System (GPS) consists of 32 navigation satellites with unique PRNs. These satellites have a medium earth orbit (MEO) with an orbital period of about 12 hours.

Broadcast ephemerides (BE) are transmitted by GPS satellites usually every 30 seconds and updated every two hours. This means that the broadcast data is valid for a time period of $t_{\text{toe}} \pm 2$ hours and outside of this time interval the broadcast shouldn't be used.

Historical broadcast ephemeris data is available in Receiver Independent Exchange

Format (RINEX) from International GNSS Service (IGS) [3]. Precise ephemeris positions that are also available from IGS.

3.4.2 BeiDou

BeiDou Navigation Satellite System consist currently of 21 navigation satellites in orbit. Unlike GPS, BeiDou has satellites of different orbit types with different orbital periods. 14 of 21 satellites have broadcast and precise ephemerides available from Multi-GNSS Experiment (MGEX) and IGS [3, 4]. BeiDou's global satellite navigation system is still in progress and will consist of 35 satellites when in full operational capacity..

BeiDou satellites PRN 1 to 5 have a geostationary orbit (GEO). As the name suggests GEO satellites are stationary over some specific point over the Earth's equator and thus have an orbital period of 24 hours.

PRN 6 to 10 have an inclined geosynchoronous orbit (IGSO). These satellites are similar to GEO satellites, but the orbital plane of the satellite is tipped some degrees such that the orbit is a stationary ellipse. The orbital period is also 24 hours.

PRN 11 to 14 are medium orbit satellites. Unlike GPS MEO satellites, BeiDou MEO satellites have an orbital period of about 13 hours.

3.5 Error Analysis

We want to know how does our prediction compare to real positions. We can compare our results to positions received from broadcast ephemeris. But, as discussed previously, broadcast reference orbit can be inaccurate. Thus we will compare our results to precise ephemeris (PE). PE positions are available much later compared to BE, which is one reason why it isn't used to predict the satellite position. But we can use PE as reference orbit for error analysis.

Broadcast and precise positions are in ECEF coordinate system and since our prediction is in Earth-centered inertial (ECI) frame, we need to change reference positions to same inertial system. After the change in the coordinate system, we can compare our predictions to reference positions, but we want to use RTN coordinate system for the comparison so we need another coordinate transformation.

Coordinate transform of vector $\mathbf{y}_{ECI}$, in a certain time-step, with positions $\mathbf{r}$ and velocities $\mathbf{v}$ from inertial frame to RTN-frame is defined as

$$\mathbf{y}_{RTN} = \begin{bmatrix} \mathbf{e}_{\mathrm{R}}^T & \mathbf{e}_{\mathrm{T}}^T & \mathbf{e}_{\mathrm{N}}^T \end{bmatrix} \mathbf{y}_{ECI} = \mathrm{R}_{RTN}\mathbf{y}_{ECI}, \tag{3.8}$$

with

$$\mathbf{e}_{\mathrm{R}} = \frac{\mathbf{r}}{\|\mathbf{r}\|}, \quad \mathbf{e}_{\mathrm{N}} = \frac{\mathbf{r} \times \mathbf{v}}{\|\mathbf{r} \times \mathbf{v}\|}, \quad \text{and} \quad \mathbf{e}_{\mathrm{T}} = \mathbf{e}_{\mathrm{R}} \times \mathbf{e}_{\mathrm{N}}, \tag{3.9}$$

where $\mathbf{y}_{RTN}$ is coordinate vector in RTN-frame, $\mathbf{e}_{\mathrm{R}}$, $\mathbf{e}_{\mathrm{T}}$ and $\mathbf{e}_{\mathrm{N}}$ are unit vectors respectively in radial, tangential and normal directions.

Since errors in radial, tangential and normal directions affect user's positioning error different amounts, we will use a metric that weights each direction according to the contribution of that direction. Such error metric is called Signal-In-Space-Range-Error (SISRE). SISRE usually takes clock error also in account, but in this work we are only interested in position error and thus we will use orbit-only SISRE. Orbit-only SISRE will be called SISRE in this work. SISRE is defined as follows

$$\mathrm{SISRE} = \sqrt{w_R^2 \Delta R^2 + w_{T,N}^2 (\Delta T^2 + \Delta N^2)}, \tag{3.10}$$

where ΔR,ΔT and ΔN are error components in radial, transverse and tangential directions, respectively. w_R and $w_{T,N}$ are weight parameters that depend on the satellite constellation. These weights are shown in Table 3.1.

	GPS	BDS GEO/IGSO	BDS MEO
w_R	0.98	0.99	0.98
$w_{T,N}^2$	$\frac{1}{49}$	$\frac{1}{126}$	$\frac{1}{54}$

Table 3.1 *SISRE weight values for different constellations. Note that the radial weight w_R is not squared, but weight in tangential and normal directions $w_{T,N}$ is. [20]*

Our prediction algorithm can fail sometimes, for example when a satellite's trajectory is changed. Thus use of rms error may not be the best choice to measure our errors. SISRE being a non-negative metric allows us to use error quantiles instead of rms error to study the errors. We mainly use 68% and 95% quantiles in our study which roughly represent one and two standard deviations σ if the errors are assumed to be normally distributed.

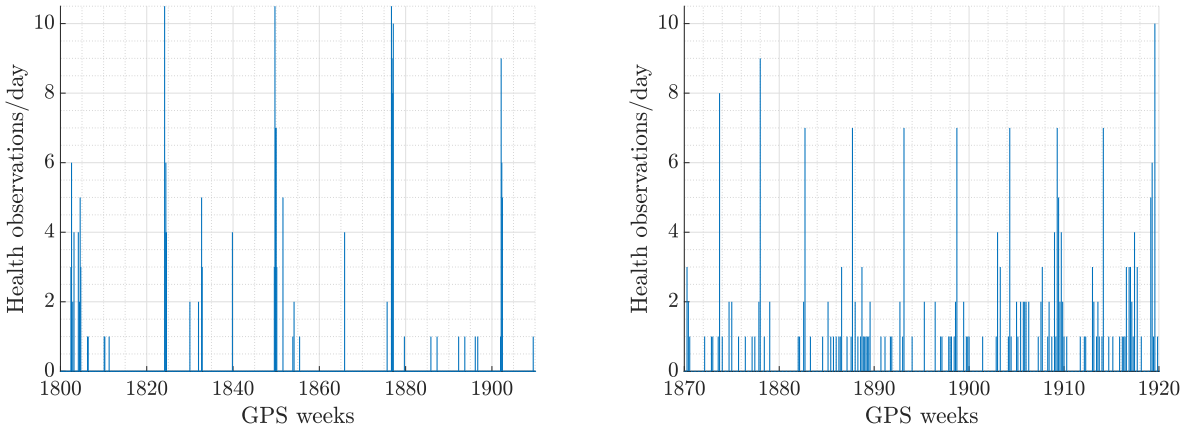
4. SATELLITE HEALTH PREDICTION

In this chapter, we will introduce satellite health prediction. Satellite health is crucial part of satellite orbit prediction since health parameter in broadcast ephemeris (BE) tells if the data sent from a satellite is unreliable. Knowing these outages beforehand would make predictions more reliable. Depending on the satellite constellation, health parameter is one or higher when the data is unreliable and zero when the data is usable. [6]

4.1 Periodicity of health parameters

Some constellations have notice advisory systems which tell users beforehand when the data sent from satellites will be unusable [1]. BeiDou doesn't have such system, which makes orbit prediction for longer periods unreliable. If we know the upcoming unhealthy period we could predict the orbit until the start of unhealthy period and stop for the duration of the period.

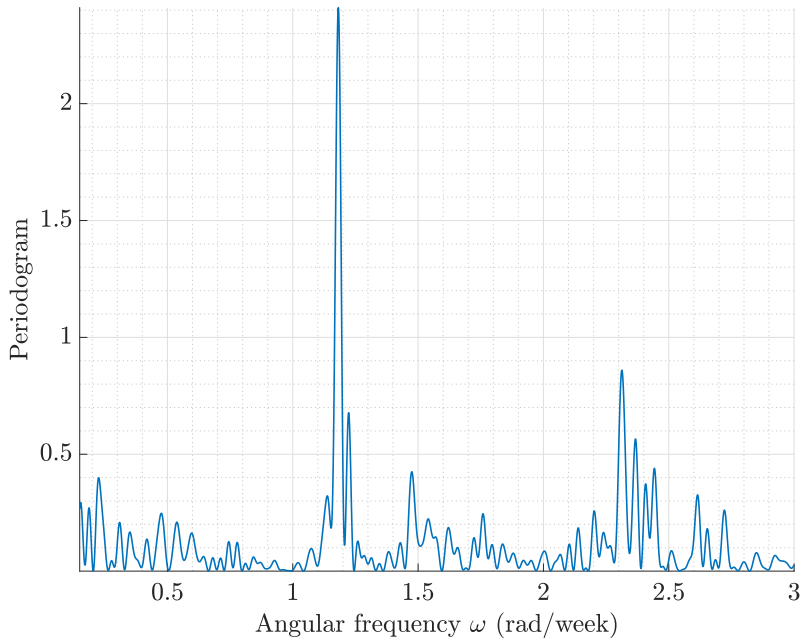
Figure 4.1 *Daily health statistics of BeiDou GEO and IGSO satellites.*



Statistics of health parameter are made by collecting the parameter from RINEX files over the period of a few years. By looking at these statistics of BeiDou GEO and IGSO satellites a period can be observed. Sum of the daily health parameters is calculated and plotted against time of observation. In Figure 4.1 the periodicity

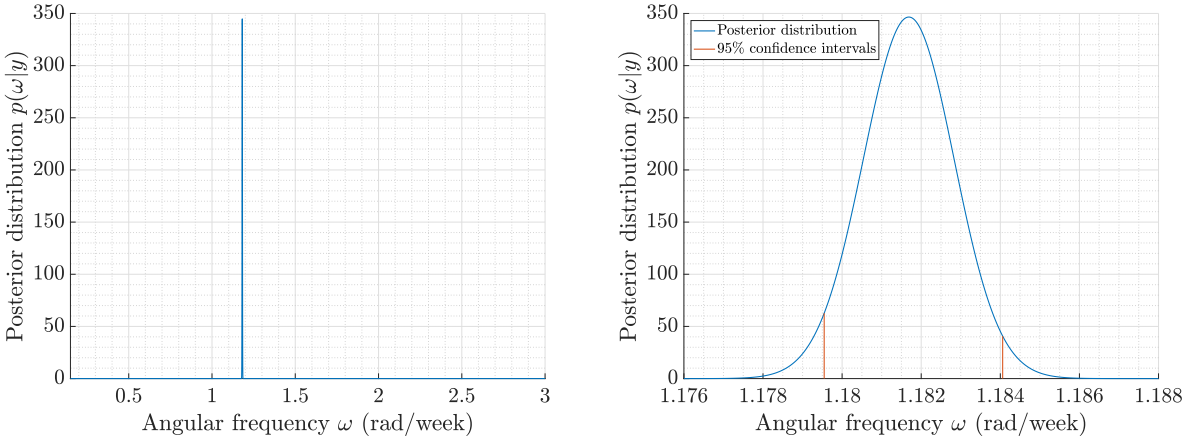
of the health parameter can be seen. Thus prediction of health parameter seems possible and, if successful, will make orbit prediction more reliable. Thus we are trying to predict the period of health parameter in weeks from the observations which are the health parameters collected from RINEX files. Periodogram is used for its Bayesian interpretation.

Figure 4.2 *Periodogram of health statistics of GPS PRN 5.*



Since the periodograms show a clear spike in the frequency, a posterior is computed with Equation (2.17). The MAP estimates can then be acquired. For GEO satellites the periods are close to five weeks and for IGSO satellites the periods are close to half a year. The posterior can be approximated to be normally distributed and confidence intervals for the estimate can be calculated. The spike shown in Figure 4.3 is really narrow, which makes the 95% confidence intervals of the estimate quite accurate.

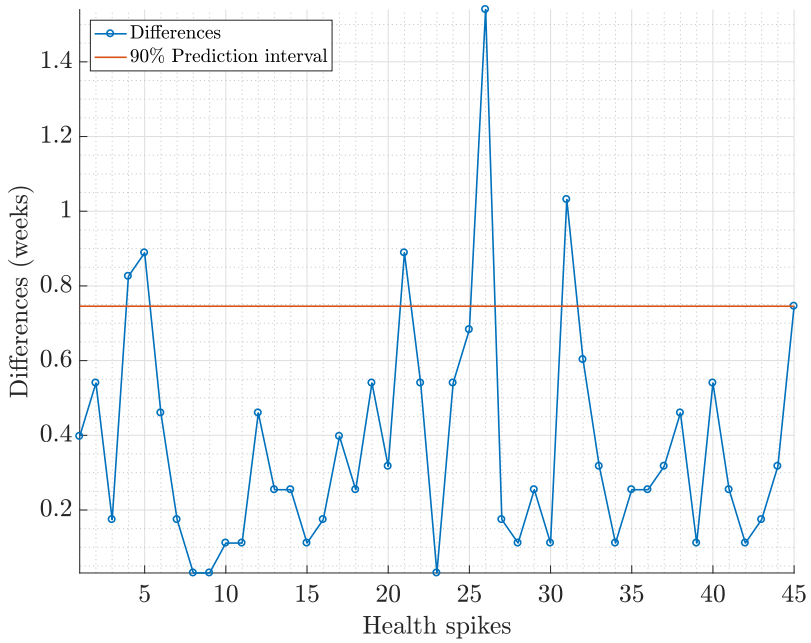
Figure 4.3 *Posterior distribution of periodogram of health statistics of GPS PRN 5. On the left is the posterior with whole observation interval shown. On the right is zoomed in version of the image on the left with confidence intervals shown. The confidence interval is* 1.818 ± 0.023 *rad/week for PRN 5.*



The period between larger health spikes is not constant, thus we need prediction intervals for the estimates. The confidence intervals are too small to be used for prediction intervals. Here we take a statistical approach to solve the intervals. Period is estimated from the periodogram and health predictions are made from larger health spikes.

Firstly we shall compute differences between predictions and real observations. Since we want the interval to be fairly small we want to reject largest differences. These are caused by misses and false alarms on the prediction. We choose the interval to include 90% of computed differences from history. This way the deviation of the period is taken into account.

Figure 4.4 *Differences of health estimates and closest health spikes matching the algorithm criteria of GPS PRN 5.*



4.2 Prediction algorithm

Since a periodicity is observed in the health parameters we can predict the next unhealthy time period assuming the period doesn't change. With the knowledge of previous unhealthy time period we can predict the next unhealthy time to be after a period determined from Equation (2.18). Taking deviation of the period into account we also add prediction intervals for the period. To avoid false positives with the algorithm a certain number of health parameters need to be observed depending on the satellite's orbit type. For GEO satellites with shorter periods 7 health counts is chosen and for IGSO satellites with longer period 6 health counts is chosen.

Algorithm 4.2.1 *Health prediction algorithm*

1. *Observation of daily health parameters. Begin prediction if*
   (a) *GEO sum* $\geq$ *7.*
   (b) *IGSO sum* $\geq$ *6.*
2. *Choose period and prediction intervals according to the PRN of the satellite.*
3. *Update period and prediction intervals if necessary.*

Also satellites β -angle can be important for IGSO and MEO satellites. When the β-angle is close to zero, more health parameters can be observed in IGSO satellites. For MEO satellites, even though no health observations are made, larger RTN and SISRE errors are observed during times of small β-angles. When β -angle is near zero the satellite is between the earth and the sun and solar radiation pressure is at its highest. During these times the predictions can be stopped for the corresponding satellite for a few weeks.

4.3 Results for health prediction

Predicted periods and deviations used in results are computed from GPS weeks 1730 to 1930. Data was collected from RINEX files and Bayesian periodogram was used for that data. Prediction periods and intervals from periodogram are listed in the Table 4.1.

Table 4.1 *Presented are predicted periods and intervals of BeiDou GEO and IGSO satellites for health prediction. PRN 1 to 5 are GEO satellites and PRN 6 to 10 are IGSO.*

PRN		Period (weeks)	Interval (± weeks)
GEO	1	4.04	0.815
	2	4.05	1.77
	3	3.76	0.903
	4	10.4	3.71
	5	5.32	0.826
IGSO	6	26.4	2.53
	7	25.6	1.20
	8	27.5	1.09
	9	25.9	2.58
	10	24.8	2.23

Most of the results were as expected. GEO satellite periods were 4-6 weeks, as expected from the preliminary tests, and intervals were small enough to allow periods long enough for orbit prediction. Period for satellite PRN 4 was observed from data to be 9 to 14 weeks and thus results are not expected to be good for deviation of five weeks. As expected, interval from our method is nearly 4 weeks and health prediction should not be used for PRN 4. For IGSO satellites, periods were expected to be around 26 weeks and thus results proved to be good. Intervals for IGSO were a generally higher than GEO intervals, which is good, since satellites with inclined geosynchronous orbit have shown higher prediction errors when β-angle is near zero [17].

Since estimated periods are different for GEO and IGSO satellites different durations for testing are chosen. For GEO satellites testing duration is chosen to be 30 weeks

from start of the year 2017, which is GPS weeks from week 1930 to 1960. For IGSO high error counts are biannual and thus testing duration is chosen to be GPS weeks 1900 to 1960.

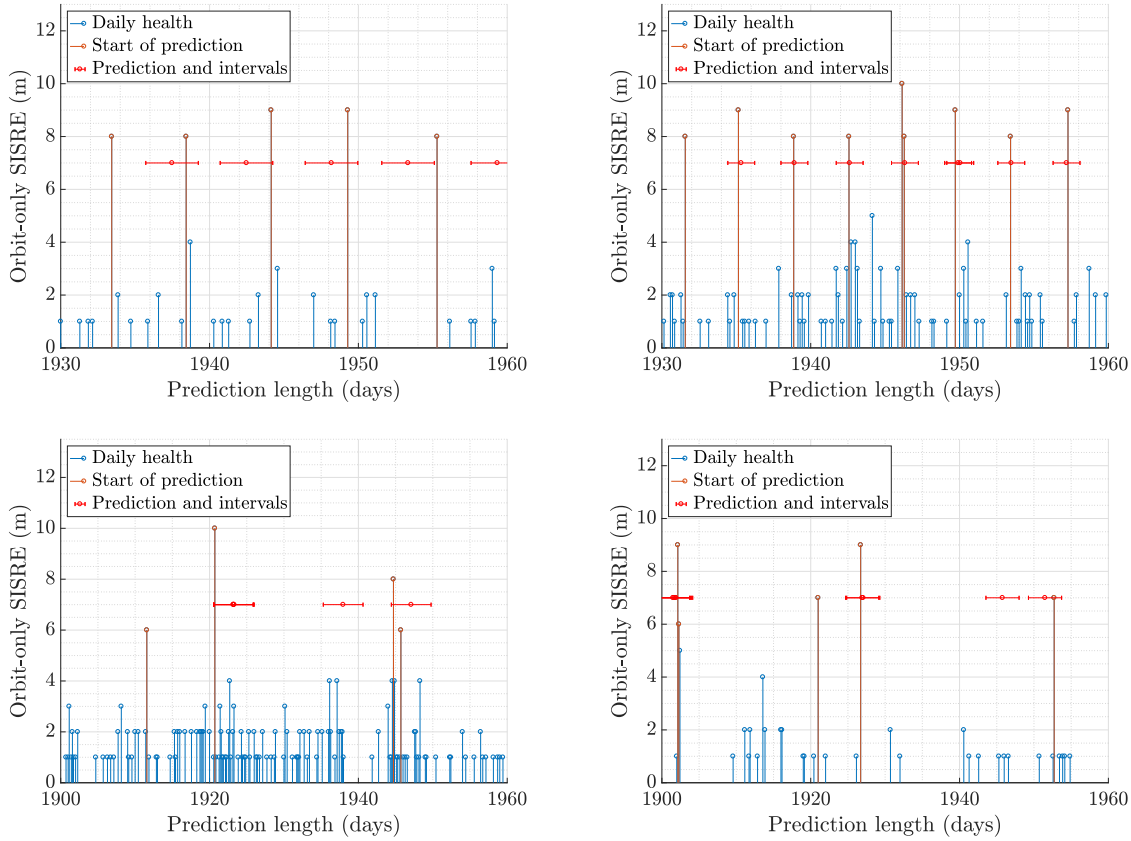


Figure 4.5 *Results of health prediction algorithm for GEO and IGSO satellites. For IGSO satellites previous prediction start is taken into account to show the results. Presented are best and worst results for both satellite types. On the left are worst results and on the right are the best results. Satellite PRNs are: top-left:2, top-right:3, bottom-left:6 and bottom-right:8*

As seen from Figure 4.5, health prediction works for both GEO and IGSO satellites. For PRN 2 results are worst for GEO, if PRN 4 is not taken into account. The period for PRN 2 seems to change during GPS weeks 1890 to 1920, which causes wrong period for prediction. For PRN 6 estimated period seems to be a bit high which shows from health bits being in left side of the interval.

Table 4.2 *Health prediction results for GEO and IGSO. Correct predictions and false alarms have been counted together by satellite type. Both IGSO and GEO have a majority in the correct predictions.*

Orbit type	Predicted healths / Percentage	False alarms/ Percentage
GEO	22 / 85 %	4 / 15%
IGSO	7 / 70 %	3 / 30%

In Table 4.2 are presented correct predictions and false alarms for GEO and IGSO. False alarms are incorrectly predicted unhealthy time intervals. The prediction duration for IGSO satellites seem to have been too short for biannual periods, since not that many health parameters was observed during testing.

5. ORBIT PREDICTION WITH DEEP LEARNING

In this chapter I will describe our method to improve our orbit predictions with convolutional neural networks and also compare the performance to previously tested methods such as latent force models [27]. As mentioned before in chapter 2, CNNs have been used successfully to learn from signals [26, 34]. This is what we also attempted with success.

Path from first to final tests are described, explaining what was done to improve the results. Choice of network layers and parameters, data scaling and modification will be explained in detail.

Two different approaches for orbit prediction are described in more detail and results will be shown for a few satellites. First method is to improve an existing prediction by estimating future errors from errors from the start of the prediction. Other method estimates the satellite position backwards in time and uses errors from that data to predict the future data. Collection of all results for the tested satellites will be listed in the appendix A. Satellites were chosen for each constellation based on typical behavior of the predictions.

5.1 Prediction of RTN errors

Radial, transverse and normal errors from predictions compared to precise or broadcast positions are made into images by compiling arrays of the errors. Single orbit prediction is used to make a single image. A two week prediction with precise reference orbit gives us a RTN error matrix of size 3×1345 with values in the first column being the estimated initial states and the remaining columns are errors in 15 minute time intervals. The input image is formed from RTN errors received from BE from a duration of four days. The input size is thus 3×384. The image with input and response is then of size 3×1729. The images are formed to have a periodic look from left to right and to achieve this we reshape the errors to a RGB image of size $48 \times 36 \times 3$. To achieve this size we need to shorten the matrix with one

value, which we choose to be the first one since it's already estimated with filtering techniques.

In Figure 5.1 is shown errors of a single orbit prediction and the image formed from the same errors.

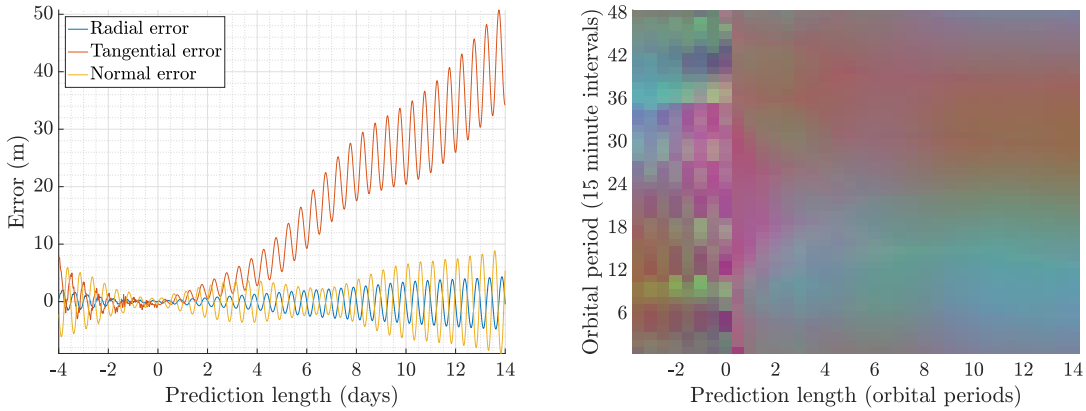


Figure 5.1 *A single orbit prediction of a GPS PRN 23. On the left is shown the RTN errors of the prediction with PE as comparison. Negative time on the prediction length axis represents RTN errors with BE as comparison with backwards propagation of the satellite. On the right is the image presentation of the same RTN errors. Each column represents a 12 hour period of the satellite prediction. First eight columns are the inputs to the convolutional layer and the rest are the response in image form.*

Computed RTN errors are placed into the channels corresponding color image's RGB values. We preprocess the data by scaling the values to a range from 0 to 1. Since we do not know the upcoming RTN errors in a real case we choose the scaling parameters to be the maximum and minimum values of all predictions for all directions. Different methods for scaling were also tested, but we chose maximum and minimum values for radial and tangential errors for each discrete time value. Normal errors maxima and minima were chosen columnwise to lessen the number of parameters. Columnwise maxima and minima were considered and tested for radial and tangential errors too, but more scaling parameters gave better results. Thus for a two week prediction we have $2 \times 1344 = 4032$ scaling parameters for radial and tangential directions and $2 \times 36 = 72$ parameters for normal direction.

Another way we used to process the images was making tangential error more constant. Since the error curve of tangential component seems to increase or decrease linearly we want to make the curve more constant. By taking the differences of successive points we get a curve that is much more constant and zero-centered. The tangential difference ΔT can be defined by

$$\Delta T_i = T_i - T_{i-1}, \quad i = 2, 3, 4, ..., n, \tag{5.1}$$

where ΔT_i is the ith tangential difference, T_i is the ith tangential error and n is the length of the error vector.

We can use the error points of the initial state we removed from the image during re-sizing to achieve this without loss of information. With the tangential difference we need to compute the scaling parameters again. The scaling parameters with tangential difference are plotted in the Figure 5.2.

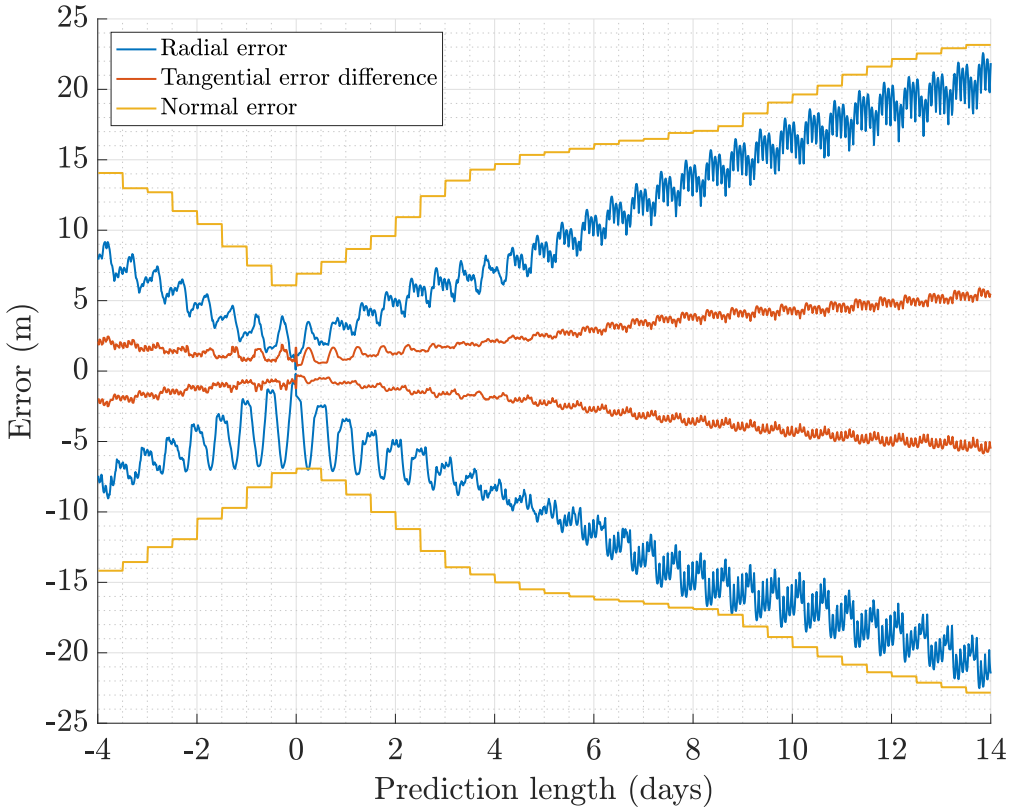


Figure 5.2 *Maximum and minimum values of RTN errors plotted as a function of prediction length. Maximum and minimum values drift away from zero the further from origin we predict the orbit.*

The images are divided into training and testing sets, both of which consist of input and response images. Training set's images are chosen randomly from all satellites to consist of two thirds of all the predictions and testing set the last third. Input is chosen to be image of the prediction of desired length and as the response we choose a two week prediction period. Since the output of the networks last layer is a vector, the response RTN errors were reshaped to be one vector of length 4032. Images with

missing data are removed, which are basically predictions during unhealthy periods. After reshaping response vectors $\mathbf{y}_{\text{response}}$ are

$$\mathbf{y}_{\text{response}} = \begin{bmatrix} \mathbf{y}_{\text{R}} & \mathbf{y}_{\text{T}} & \mathbf{y}_{\text{N}} \end{bmatrix}, \tag{5.2}$$

where $\mathbf{y}_{\text{R}}$, $\mathbf{y}_{\text{T}}$ and $\mathbf{y}_{\text{N}}$ are vectors containing radial, tangential and normal errors respectively.

The images are then passed through the convolutional neural network (CNN) with layers fitted for regression. Since the images are already small a simple network architecture was enough. Pooling also wasn't used since number of parameters predicted is higher than parameters in images, thus information-loss from pooling is not wanted. Learn rate and momentum were chosen to be 0.001 and 0.9 respectively.

Layers were chosen to be quite simple for the convolutional neural network. First layer is the image input layer, which zero-center normalizes the data if the data isn't already normalized. Our scaling was done between 0 and 1 and thus the network zero-center normalizes the data.

Second layer is convolution layer with convolutions same size as the image width. The number of convolution filters was chosen to be eight, but higher and smaller number of filters were tested. With fewer filters the results were computationally faster but worse in prediction accuracy and with more filters the computation time was higher with smaller improvement. The output of the convolutional layer is thus a vector of length 41 for each filter. Thus the RTN errors are connected in each filter.

After convolutional layer the data is non-linearized in the rectified-linear-unit layer, which turns negative values from convolutional layers output to zero. The output size is not affected in the ReLU-layer.

Now we fully connect the output in the fully connected layer. The outputs of each filters are put together and multiplied with a weight matrix. When our eight filters are put together the size changes to a vector with length 328, which is 41 times 8. Since we want predictions for two weeks, with 15 minute interval, our weight matrix is of size 4032×328. Fully connected layer can be represented as Equation 2.29. The output size is now a vector of length 4032, which represents predicted radial, tangential and normal error vectors as shown in Equation (5.2).

In the regression layer, mean-squared error of the predicted outcome and response is calculated. This error is minimized with backpropagation and thus this layer is

not used when predicting with the CNN.

After the training, network is tested with the testing set. The output is a prediction of RTN error for the two week prediction. After re-scaling the data to real values, using the same maxima and minima parameters from scaling and inverting the tangential difference operation, results are compared with real RTN errors. The RMSE between real and predicted RTN errors were computed.

Training set was also used in testing the neural network to prevent overfitting. Results were nearly the same with training and testing set and thus overfitting was not a problem. Shown in Figure 5.3 is the RMSE of training and test image sets. The RMSE of training is barely different compared to RMSE of test set as seen from figure 5.3.

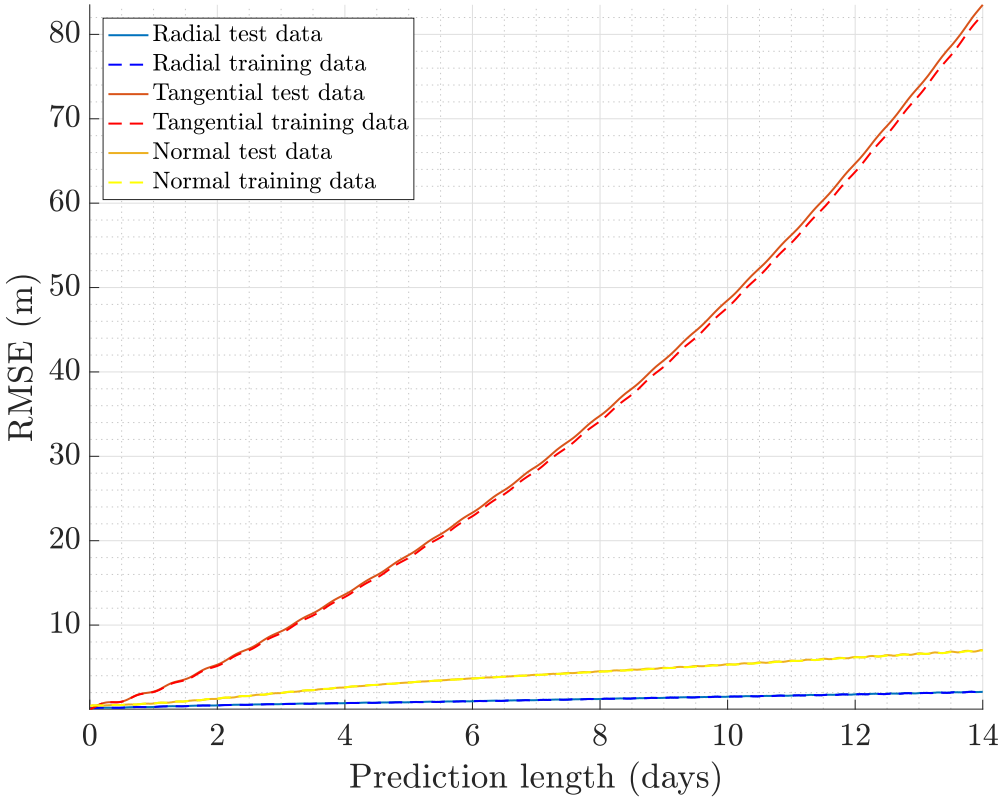


Figure 5.3 *RMSE of the network training and test data sets as a function of time. The test data performed almost the same as the training data, which means overfitting didn't happen. Highest tangential errors in training and testing were about 1000 meters and thus tangential errors RMSE of 80 meters is acceptable.*

To improve predictions from the network, many things can be changed. Firstly we can change and fine-tune our layers in the network. Secondly we can change the

sizes of the images. And thirdly we can process the data in the images themselves as we did with the tangential errors. Without the tangential difference the network still works with no large effects in accuracy, but the curves are really noisy. In Figure 5.4 the noisy output of the tangential error can be seen.

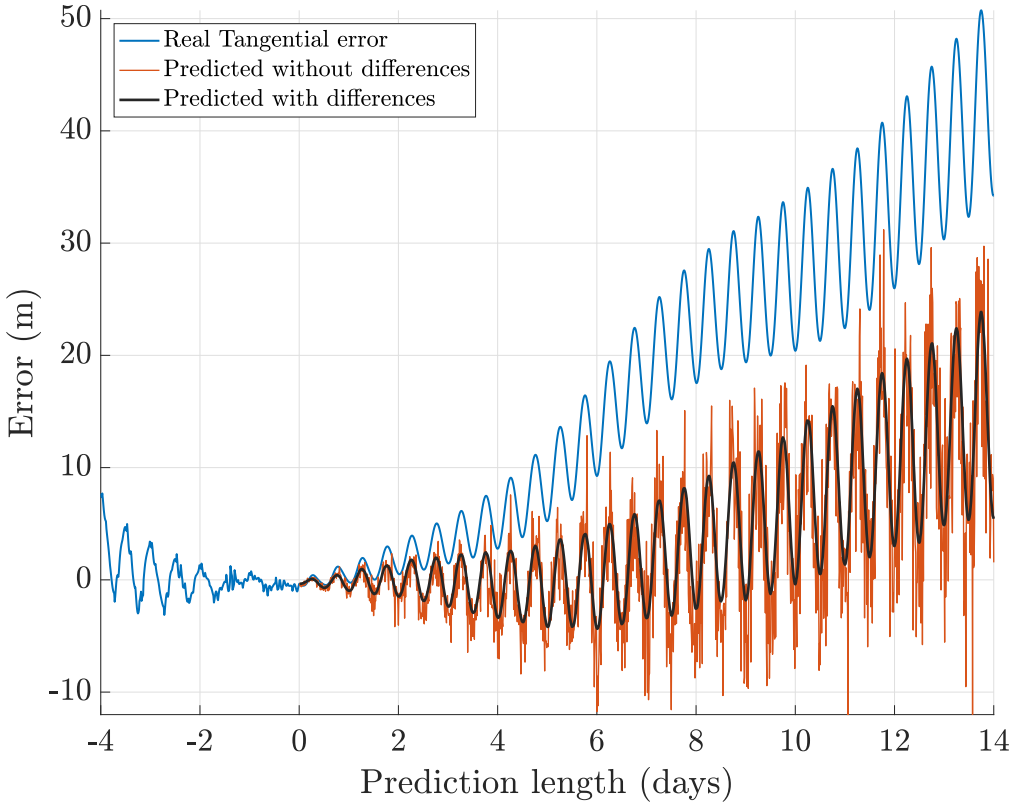


Figure 5.4 *Comparison of predicted tangential error with and without the tangential difference processing.*

To make error curve prediction more practical and useful for applications, a backwards method was tested. Backwards method means that we predict the satellite orbit backwards in time, since we already have the broadcast data available. Then we proceed to predict the error curves of our orbit prediction forward in time. Both methods give similar results in testing period and thus both methods were further tested to improve our orbit predictions. In Figure 5.5 the results of both methods for the same orbit prediction can be seen.

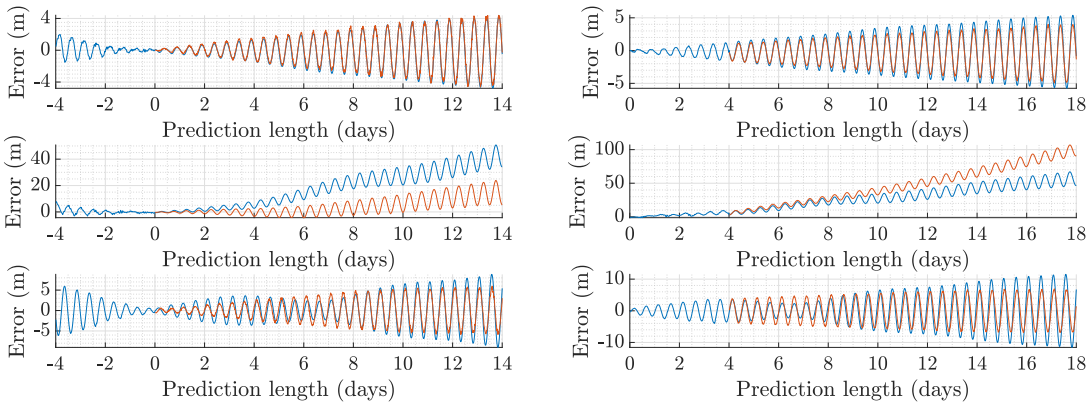


Figure 5.5 *A single orbit prediction of a GPS PRN 23. Directional errors are from top to bottom radial, tangential and normal errors. With blue are the original RTN errors and with red are the predicted RTN errors. On the left is shown the backward method for predicting the upcoming errors and on the right is the forward method. Both methods seem to work fairly well but the radial error having highest differences.*

For BeiDou's GEO and IGSO satellites we shall choose the images to have width of 18, thus each column represents prediction errors of a day. Thus we reshape the image from size 48×36 to 96×18. The image looks more periodic as seen from Figure 5.6.

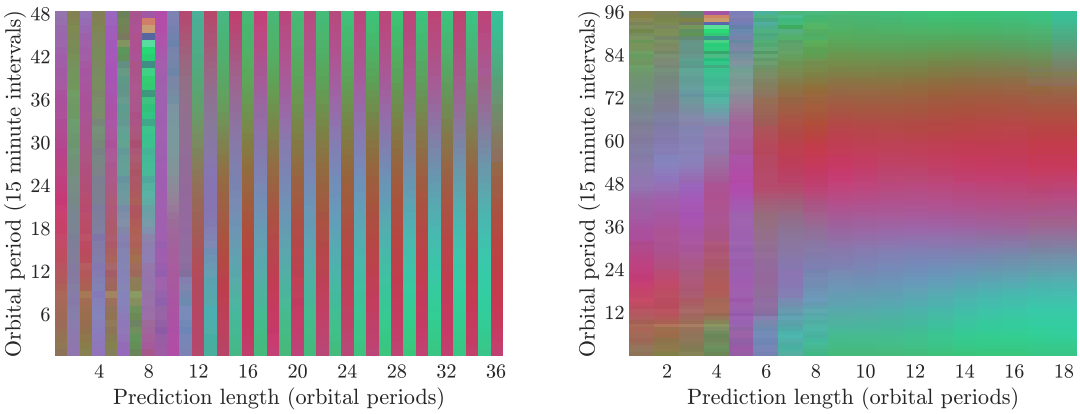


Figure 5.6 *A single orbit image for BeiDou GEO satellite. On the left the image is of size* 48×36 *and on the right the size is* 96×18

The orbital period of GEO and IGSO satellites is 24 hours and thus images whose column is a full days data is more periodic from left to right. With the changed image size the convolutional layers filter size is also changed to 4×4.

5.2 Using CNN for orbit prediction

When we have trained the CNN for a constellation we can use it to improve our predictions. In forward CNN method we start the prediction as before, but after few days of prediction we compute RTN error curves from BE orbit. Then we predict the upcoming RTN errors from the computed data and correct our orbit predictions according to our error predictions. In backward CNN method we predict the orbit backwards for a few days and compute the RTN errors. Similarly to forward method we then predict the upcoming RTN errors and correct our orbit prediction with the error predictions.

Now we can process the RTN data to be used in the CNN. We scale the data, perform tangential difference and reshape prediction curve to an image. Now we can predict the errors for the rest of the prediction. We get the prediction vector $\hat{\mathbf{y}}_{RTN}$ as an output of the CNN. To use the error predictions to improve our orbit prediction we need to change the coordinate system of predicted errors from RTN back to inertial frame. We can change the frame using the transformation matrix R_{RTN} from Equation (3.8).

$$\hat{\mathbf{y}}_{RTN} = \mathrm{R}_{RTN}^{-1}\hat{\mathbf{y}}_{\mathrm{ECI}}, \tag{5.3}$$

where $\hat{\mathbf{y}}_{\mathrm{ECI}}$ is the error predictions in the inertial frame.

Now we can simply reduce our correction parameters from our orbit prediction. To analyze our prediction accuracy we change our orbit predictions coordinate frame back to RTN.

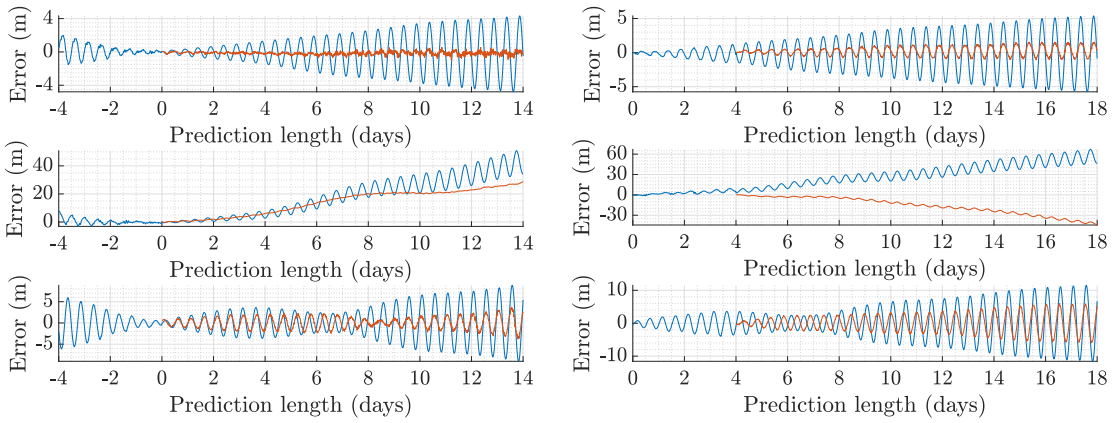


Figure 5.7 *A single orbit prediction of a GPS PRN 23 with corrected RTN errors. Directional errors are from top to bottom radial, tangential and normal errors. With blue are the original RTN errors and with red are the RTN errors with correction from the CNN. On the left is shown the backward method and on the right is the forward method.*

In the Figure 5.7 is presented an example of the error corrections made with CNN with forward and backward methods.

5.3 Results for GPS

First we shall take a look at results for GPS satellites. Accuracy was improved for the regular orbit prediction for all GPS satellites. For most GPS satellites backward CNN method improved better or roughly the same as forward method. More complete results for all satellites are presented in Table 5.1. Predictions for training the CNN were made from GPS weeks 1835 to 1870 with around 3000 predictions per satellite. After cleaning up the data from orbit predictions that failed or had missing data, a CNN was trained for all the satellites. The CNN and scaling parameters were then taken and used to make more predictions. Twenty week period of GPS weeks 1887 to 1907 was chosen to make new predictions using the CNN.

Both forward and backward time CNN methods were tested and are presented in Figure 5.8. We will take a look at the 68% and 95% quantiles of the predictions made during the testing period. The results are compared to predictions made without the CNN and using a previous model used in our research group, latent force model (LFM) [27]. Latent force models testing is made differently compared to CNN predictions. CNN is tested systematically and LFM is tested by simulating real prediction process. Predictions for LFM are continued with new BEs with random prediction lengths for 20 weeks period. Failed predictions with CNN and LFM, caused by for example scheduled correction of orbit, were removed from the analysis to make quantiles more reliable. The number of removed predictions is less than 5% percent of all predictions per satellite. Also if any of the predicted errors from CNN was larger than 1000 meters, the CNN prediction was discarded as failed and no error corrections were made to the orbit prediction.

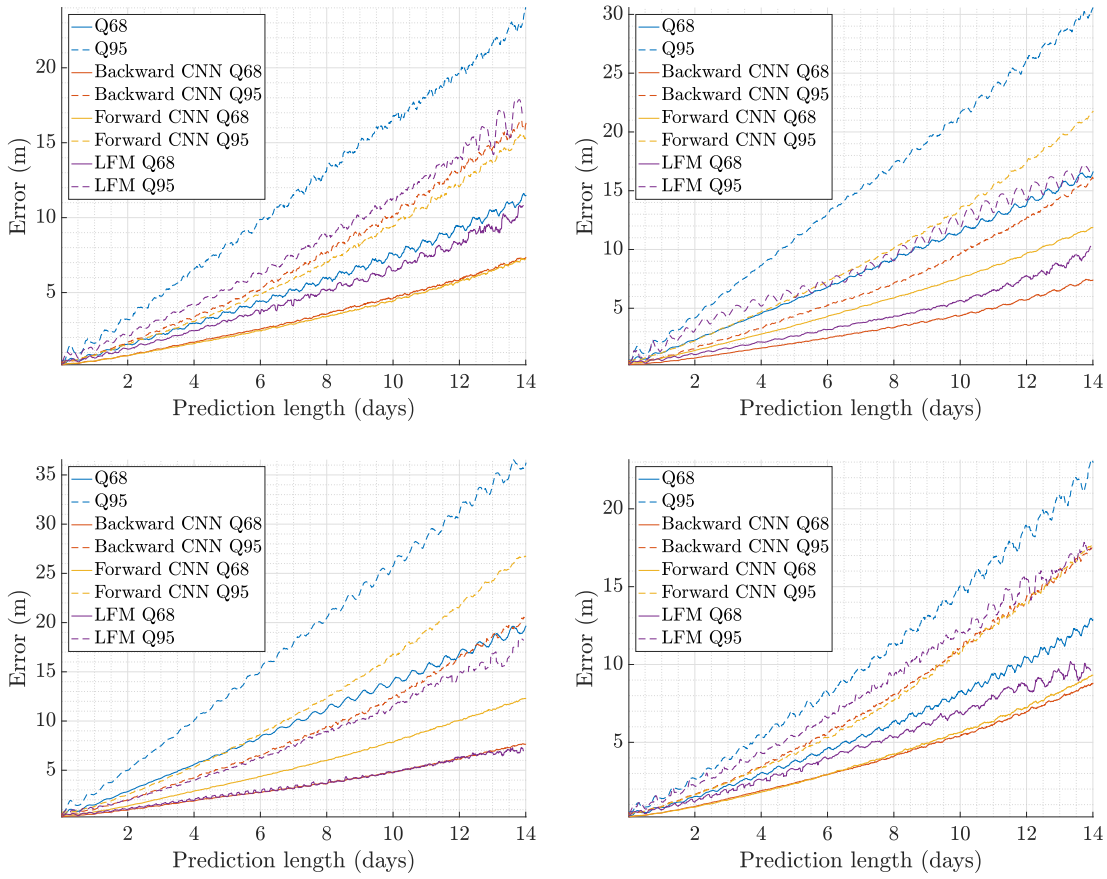


Figure 5.8 *Results of both CNN prediction methods for GPS satellites with PE and LFM as comparison. Both CNN methods improve prediction accuracy as seen from figures. Satellite PRNs are: top-left:5, top-right:6, bottom-left:8 and bottom-right:15*

Results show that the error correction method with CNN works and improves the orbit prediction quantiles. Of course results vary by satellite, but all GPS satellites show improvements. Smallest improvement of 14 days prediction 68% quantile in percentage was shown in GPS PRN 14, which still improved by 17.5%, when largest improvements were in GPS PRN 26 which showed 68.3% improvement. These results are significant and I expect that, with further tuning, they could still be improved further. Compared to LFM, backward CNN method works well.

In Table 5.3 the complete results for GPS are presented. Missing results for few satellites are caused by change in satellite and unhealthy periods for the satellite.

5.4 Results for BeiDou

Now we shall take a look at results for all three BeiDou satellite types. Orbit predictions are compared to regular predictions without CNN and LFM methods. Since

Table 5.1 *Complete results of CNN prediction for GPS satellites. SISRE column means orbit-only SISRE quantile accuracy in meters. Improvement column means orbit-only SISRE quantile accuracy change in percents when comparing predictions with and without CNN method. Results are rounded to three significant digits. In improvement percentages are shortened with I-%.*

GPS

PRN	7 days prediction				14 days prediction			
	Q68%		Q95%		Q68%		Q95%	
	SISRE (m)	I-%	SISRE (m)	I-%	SISRE (m)	I-%	SISRE (m)	I-%
1	3.11	37.1	6.01	47.6	6.95	32.5	14.4	36.9
2	3.55	44.7	6.18	54.7	10.5	31.3	18.4	35.9
3	3.06	58.6	8.26	52.0	7.39	50.0	19.9	43.0
4	-	-	-	-	-	-	-	-
5	3.19	37.9	6.76	41.4	7.58	34.1	17.2	28.5
6	2.91	63.6	6.15	59.5	7.43	55.3	16.0	47.7
7	3.24	45.1	6.29	49.2	8.15	36.5	17.8	34.8
8	3.21	67.6	7.97	55.8	7.63	61.2	20.5	43.5
9	3.44	62.9	7.42	57.4	8.21	57.6	19.9	45.9
10	3.63	55.4	10.3	50.5	8.93	45.9	29.1	39.6
11	4.70	29.5	8.76	35.9	13.7	19.7	23.6	27.5
12	4.14	36.3	9.14	35.0	12.8	19.8	23.8	26.6
13	3.64	31.0	7.55	30.0	8.15	28.3	19.7	17.9
14	3.83	30.1	8.60	20.6	10.0	17.5	22.7	7.77
15	3.55	35.0	7.00	28.0	8.82	31.4	18.3	20.4
16	3.72	45.0	9.91	32.5	8.70	38.6	25.1	20.9
17	2.91	47.9	6.24	46.1	7.67	36.9	17.4	26.9
18	4.56	33.6	8.39	31.7	13.9	26.7	25.0	20.4
19	4.21	33.3	9.55	22.1	12.2	24.8	27.9	7.05
20	3.45	40.3	6.78	39.0	8.65	33.8	17.1	31.8
21	3.65	42.1	7.01	49.6	9.01	35.0	18.7	37.5
22	4.33	29.2	7.90	35.8	11.5	23.9	19.9	26.1
23	3.93	31.7	6.86	31.6	10.2	25.5	20.1	19.0
24	3.68	50.7	7.43	47.6	9.32	39.7	21.0	32.3
25	3.04	48.6	8.49	34.2	6.92	42.9	18.5	28.3
26	3.02	70.2	8.47	58.7	6.46	68.3	22.3	47.0
27	3.32	63.0	8.47	49.1	7.17	60.8	20.2	39.6
28	2.76	54.2	7.60	44.1	6.96	45.6	21.6	26.8
29	4.05	35.5	7.49	40.3	10.6	25.4	21.0	23.6
30	-	-	-	-	-	-	-	-
31	3.47	41.9	7.12	44.3	10.0	23.7	19.3	30.4
32	-	-	-	-	-	-	-	-

backwards CNN model performed better than forward method for GPS satellites, only backwards CNN model is tested for BeiDou satellites. Training data for BeiDou was collected from the same weeks as for GPS. Also testing was done for the same weeks as GPS was tested. CNN method improved results greatly compared to regular orbit prediction. Complete results for BeiDou are listed in Table 5.3.

GEO satellites have generally shown worst results of the three orbit types. This is mainly because of GEO satellites orbital maneuvers that are made roughly every 4 to 5 weeks [33] and orbit predictions over time intervals that span these maneuvers fail. Large amounts of health parameters are observed during these maneuvers, and thus data from previous four days might not be available for CNN.

Health prediction can be used to lessen the amount of failed predictions notably. A failed prediction is a prediction which tangential error rises over 1000 meters in two week prediction. If the CNN predicts any of the errors over 1000 meters the prediction is marked as failed and ended. The CNN and health prediction lessen the number of failed predictions greatly and improve the orbit prediction for all GEO satellites. CNN also improves results of satellite PRN4 for which the health prediction doesn't work. Results for two of the GEO satellites can be seen from the Figure 5.9.

Table 5.2 *Presented are started predictions with and without health prediction in use. When health prediction is used number of started predictions is much lower, since predictions are not made during unhealthy periods. Number of failed predictions decreases greatly when using health prediction.*

PRN		Without health prediction			With health prediction		
		Started predictions	Failed predictions		Started predictions	Failed predictions	
			PE	CNN		PE	CNN
GEO	1	1735	804	1010	999	95	143
	2	1715	1078	1151	860	228	335
	3	1688	1023	1213	1011	91	88
	4	-	-	-	-	-	-
	5	1717	624	789	1143	108	107
IGSO	6	1726	198	228	1383	125	121
	7	1737	214	248	1449	121	121
	8	1752	199	198	1643	138	137
	9	1743	214	247	1366	113	113
	10	1748	213	296	1381	116	161

For IGSO satellites we also use health prediction in case of near zero β-angle. For

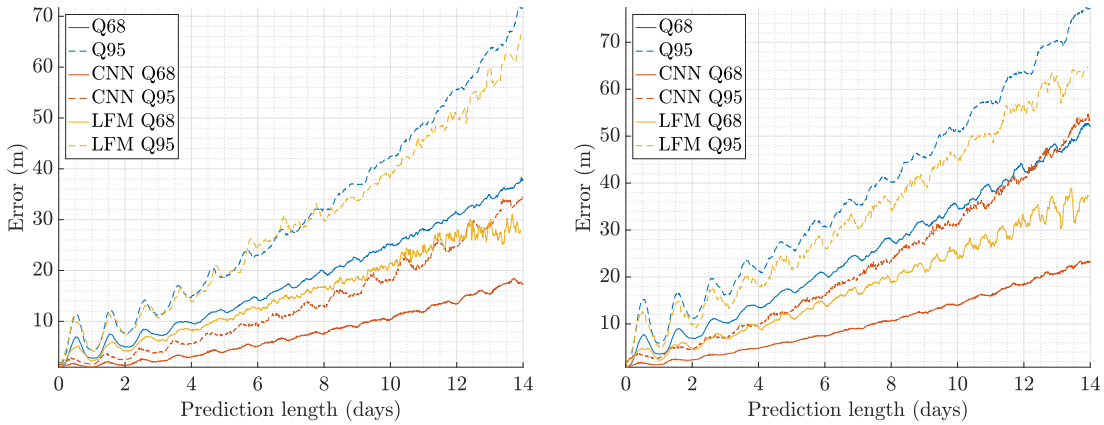


Figure 5.9 *Results of backwards CNN prediction method for BeiDou GEO satellites with prediction without corrections and LFM as comparison. Presented are worst and best results of GEO satellites. Satellite PRNs are: left:1, right:5*

all IGSO satellites unhealthy periods were predicted and during these times high errors are caused by solar radiation pressure. In the Table 5.2 is presented the effect of health prediction to the number of failed predictions. CNN method works for IGSO satellites very well and again results are also compared to LFM. Results for two IGSO satellites are shown in the Figure 5.10.

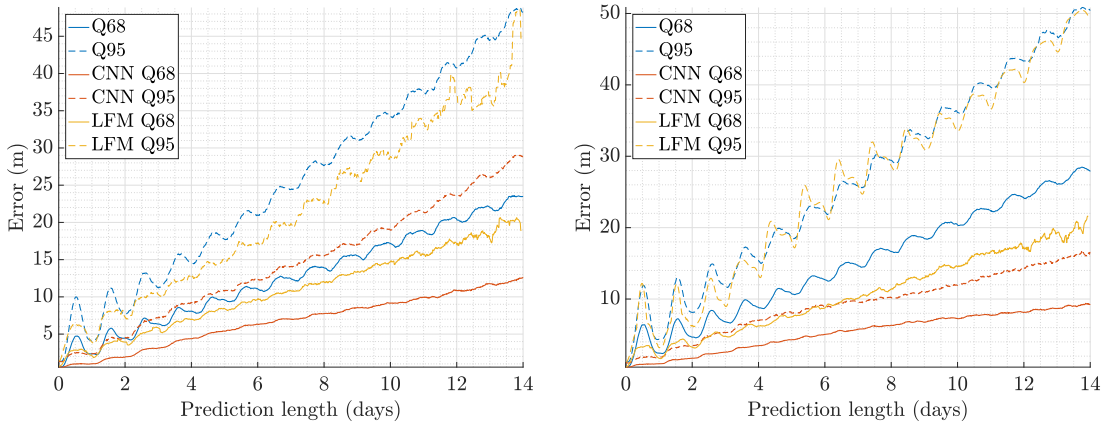


Figure 5.10 *Results of backwards CNN prediction method for BeiDou IGSO satellites with prediction without corrections and LFM as comparison. Presented are best and worst results of IGSO satellites. Satellite PRNs are: left:6, right:7*

As seen from Figures 5.9 and 5.10, CNN method works exceptionally well for these satellite orbit types. For all but one GEO satellite (PRN5) and for two IGSO satellites (PRN 7 and 10), the 95% quantile was better than regular predictions 68% quantile. Largest improvement of GEO satellites 68% quantile of 14 days prediction was 54.8% and smallest was 44.6%. For IGSO largest was 66.7 % and smallest was 46.7%. Results for GEO and IGSO satellites are really good when compared to

LFM. Especially 95% quantile shows high improvement from LFM.

Least improvement in 95% quantile with CNN was shown in MEO satellites. Orbit prediction performance was still improved for all MEO satellites with CNN method. Results for two MEO satellites are shown in the Figure 5.11. MEO improved least compared to LFM.

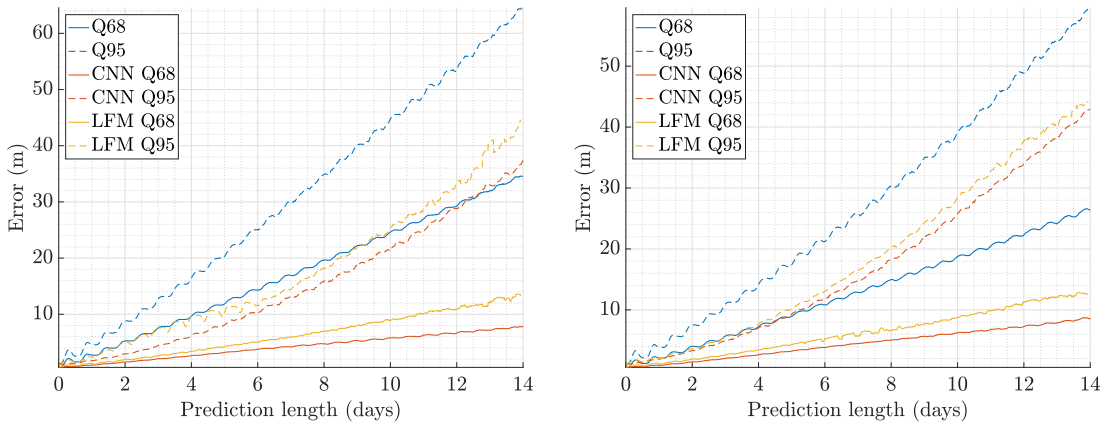


Figure 5.11 *Results of backwards CNN prediction method for BeiDou MEO satellites with prediction without corrections and LFM as comparison. Presented are best and worst results of MEO satellites. Satellite PRNs are: left:11, right:12*

The 68% quantile for MEO satellites had highest improvement in percentage of all BeiDou satellites, but improvement compared to LFM was quite small. Smallest improvement in 68 % quantile of 14 days prediction was 67.7 % and highest was 77.5 %. The 95% quantiles didn't improve as much with smallest improvement being 27.9 % and highest 41.6 %.

Overall results for BeiDou are very good with the combined health prediction and CNN prediction methods. Radial and tangential errors had highest improvement, but normal error could still be improved further with tuning of the network. Backwards CNN method works really well with BeiDou satellites with 68 % quantiles average improvement being over 50 %.

In Table 5.3 the complete results for BeiDou are presented. Missing results for PRN13 are caused by change in satellite.

Table 5.3 *Complete results of CNN prediction for BeiDou satellites. SISRE column means orbit-only SISRE quantile accuracy in meters. Improvement column means orbit-only SISRE quantile accuracy change in percents when comparing predictions with and without CNN method. Results are rounded to three significant digits. In improvement percentages are shortened with I-%.*

BEIDOU

PRN		7 days prediction				14 days prediction			
		Q68%		Q95%		Q68%		Q95%	
		SISRE (m)	I-%	SISRE (m)	I-%	SISRE (m)	I-%	SISRE (m)	I-%
GEO	1	6.41	61.8	11.2	59.7	17.6	53.7	34.8	51.4
	2	10.1	56.1	17.4	50.3	29.8	44.9	45.4	43.1
	3	8.87	55.2	15.4	53.0	23.8	44.6	38.0	49.0
	4	7.29	64.3	13.9	59.0	19.4	53.9	36.1	51.0
	5	9.05	62.5	20.0	44.0	23.4	54.8	53.4	31.0
IGSO	6	6.98	43.6	14.1	42.4	12.5	46.7	28.8	40.2
	7	5.77	60.6	9.5	63.1	9.3	66.7	16.5	67.4
	8	7.76	46.6	13.9	47.2	14.9	50.0	40.4	27.2
	9	6.36	43.3	11.8	46.1	11.2	47.1	20.1	50.8
	10	6.14	55.4	9.69	59.8	11.0	57.9	19.3	60.1
MEO	11	4.19	75.2	13.2	55.9	7.79	77.5	37.5	41.6
	12	4.48	65.2	14.8	42.6	8.51	67.7	43.0	27.9
	13	-	-	-	-	-	-	-	-
	14	5.01	68.8	22.5	28.7	9.17	70.9	51.2	28.4

6. CONCLUSIONS AND FUTURE WORK

The goal of this study was to see if deep learning and spectral analysis could be used to improve satellite orbit predictions, either in quality or accuracy. Health prediction for BeiDou satellites was introduced to improve quality of satellite predictions. The simple model we used was made using spectral analysis based on data from previous few years. convolutional neural networks with simple layers were tested to improve accuracy of orbit predictions. Errors from history data was used to predict upcoming errors and used to correct our orbit predictions.

A Bayesian interpretation of periodogram was used to find the period of health bits from broadcast data of GEO and IGSO satellites. The periods were clearly present in the periodograms and were easily used to create a simple prediction algorithm. The algorithm works for both GEO and IGSO satellites, but it also has its faults. The algorithm can't predict smaller number of health bits and larger number of health bits that occur between the regular period. The algorithm is still relatively simple and can be used easily to prevent orbit predictions during unhealthy periods.

The convolutional neural network used in this study was simple and the layers could be made much more deeper. Deeper network could possibly increase the accuracy of the results, although good results were achieved with very simple layers. Also more data could be used for the network's input images to increase accuracy. The model used in this study showed improvements beyond our expectations and proved that deep learning could be used to improve orbit accuracy. The convolutional neural networks were made by orbit type and orbital periods, thus one network could be used for many satellites in one constellation. Better results are a possibility if the networks are dedicated to one satellite only. This way insufficiency of training data could be an issue, but a network for all satellites of same type could be transfer learned to work only for one satellite using less amount of data.

If the network was either trained or transfer learned with data only few weeks older than test predictions, better results could be a possibility. Our network was trained with predictions from weeks 1835 to 1877 and tested with predictions from 1887 to 1907 thus leaving over a year between some predictions. In a year SRP parameters

need to be changed and thus shape of the RTN curves can be different. A network could be updated with transfer learning from a newer data and thus same SRP parameters could be used. When enough time has passed the network could be updated again with the predictions of past few weeks.

What was also considered was for each radial, tangential and normal errors to have their own neural networks. But in this study we decided to study them as a whole in one neural network. Since the error curves of radial and tangential directions are quite similar, the network might minimize their error more quickly and thus leaving normal error with more to improve on. A possibility of adding another channel with clock errors was also considered but left for further work.

A recurrent convolutional neural network model was also considered. We would only predict the orbit for a few days and then get new ephemeris data to correct prediction again recurrently. This method would probably get better results, but the method would also require a lot of data to work. Also a recurrent version of our network is not very applicable, since the network needs new data every few days instead of every few weeks and thus starting a new prediction would be much more simple.

We also considered the size of the convolutional neural network. By having a simple neural network with few layers and small number of filters in the convolution layer our network stays relatively small in size to be more applicable. But the number of parameters in the network is still quite large and the amount of data required to actually predict with the network is a lot.

As future work, we left other constellations to be tested with our method, since only GPS and BeiDou were tested for out methods. But since the results are good for all three orbit types BeiDou satellites, good results for other constellations are expected. Deep learning methods other than CNN are also left to future work. Recurrent neural networks and recurrent convolutional neural networks could be applied for use in improving either orbit prediction accuracy or quality. Left for future work was also the possibility of predicting errors from other methods such as LFM, to see if a combined model could improve orbit accuracy even further.

In conclusion, spectral analysis can be used to predict unhealthy periods for BeiDou satellites and deep learning methods such as convolutional neural networks could be used to improve orbit prediction accuracy and quality. Improvements can be made by predicting future errors from previous broadcast data with neural networks and these improvements can be better than improving the force model by adding more analytical terms.

BIBLIOGRAPHY

[1] "GPS NANU messages." [Online]. Available: https://celestrak.com/GPS/NANU/

[2] "IERS Earth orientation data." [Online]. Available: https://www.iers.org/IERS/EN/DataProducts/EarthOrientationData/eop.html

[3] "IGS Products." [Online]. Available: http://www.igs.org/products

[4] "MGEX products." [Online]. Available: http://mgex.igs.org/IGS_MGEX_Products.html

[5] "Global Positioning Systems Directorate Systems Engineering Integration Interface Specification IS-GPS-200," Tech. Rep., Dec. 2015.

[6] "RINEX - The Receiver Independent Exchange Format, Version 3.03," Tech. Rep., Jul. 2015.

[7] J. Ala-Luhtala, M. Seppänen, S. Ali-Löytty, R. Piché, and H. Nurminen, "Estimation of initial state and model parameters for autonomous GNSS orbit prediction," in *International Global Navigation Satellite Systems Society Symposium 2013 (IGNSS2013)*, Jul. 2013.

[8] J. Ala-Luhtala, M. Seppänen, and R. Piché, "An empirical solar radiation pressure model for autonomous GNSS orbit prediction," in *Proceedings of PLANS 2012 IEEE/ION Position Location and Navigation Symposium*, 2012.

[9] D. S. Bernstein, *Matrix Mathematics*. Princeton University Press, 2009.

[10] G. L. Bretthorst, *Bayesian Spectrum Analysis and Parameter Estimation*. Springer-Verlag, 1988.

[11] China Satellite Navigation Office, "BeiDou Navigation Satellite System Signal In Space Interface Control Document Open Service Signal (Version 2.0)," Tech. Rep., Dec. 2013.

[12] R. Fisher, "Description of JPL Solar System Ephemeris." [Online]. Available: http://www.cv.nrao.edu/~rfisher/Ephemerides/ephem_descr.html

[13] J. Hartikainen, M. Seppänen, and S. Särkkä, "State-space inference for non-linear latent force models with application to satellite orbit prediction," in *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*, Jun. 2012. [Online]. Available: http://icml.cc/2012/papers/477.pdf

[14] S. Jan and A. Tao, "The Open Service Signal in Space Navigation Data Comparison of the Global Positioning System and the BeiDou Navigation Satellite System," *Sensors (Basel, Switzerland)*, 2014.

[15] A. Karpathy, "Convolutional Neural Networks (CNNs / ConvNets)." [Online]. Available: "http://cs231n.github.io/convolutional-networks/"

[16] M. P. Knapp, "Sines and Cosines of Angles in Arithmetic Progression," *Mathematics Magazine*, pp. 371–372, Dec. 2009.

[17] H. Leppäkoski, S. Rautalin, X. Zhang, S. Ali-Löytty, and R. Piché, "Extended prediction of QZSS orbit and clock," in *2016 International Conference on Localization and GNSS*, 2016.

[18] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements, and Performance*, 2nd ed. Ganga-Jamuna Press, 2006.

[19] O. Montenbruck and E. Gill, *Satellite Orbits: Models, Methods and Applications*, 3rd ed. Springer, 2005.

[20] O. Montenbruck, P. Steigenberger, and A. Hauschild, "Broadcast versus precise ephemerides: a multi-GNSS perspective," *GPS solutions*, pp. 321–333, 2015.

[21] R. Ng, "Deep Convolutional Networks," 2017. [Online]. Available: "http://www.ritchieng.com/machine-learning/deep-learning/convs/"

[22] NGA, "EGM2008 model cofficients." [Online]. Available: http://earth-info.nga.mil/GandG/wgs84/gravitymod/egm2008/first_release.html

[23] A. Penttinen and R. Piché, *Bayesian Methods*. Tampere University of Technology, 2010. [Online]. Available: http://urn.fi/URN:NBN:fi:tty-201012161393

[24] A. Pukkila, "GNSS satelliitin radan ennustamisen virhetermien analysointi," Master's thesis, Tampere University of Technology, Jun. 2015. [Online]. Available: http://URN.fi/URN:NBN:fi:tty-201505081267

[25] A. Pukkila, J. Ala-Luhtala, R. Piché, and S. Ali-Löytty, "GNSS orbit prediction with enhanced force model," in *2015 International Conference on Localization and GNSS (ICL-GNSS)*, Jun. 2015.

[26] S. Qu, J. Li, W. Dai, and S. Das, "Understanding Audio Pattern Using Convolutional Neural Network From Raw Waveforms," Nov. 2016.

[27] S. Rautalin, "Data-driven force models in GNSS satellite orbit prediction," Master's thesis, Tampere University of Technology, 2016. [Online]. Available: http://URN.fi/URN:NBN:fi:tty-201703211195

[28] S. Rautalin, S. Ali-Löytty, and R. Piché, "Latent force models in autonomous GNSS satellite orbit prediction," in *2017 International Conference on Localization and GNSS*, 2017.

[29] B. Rohrer, "How do Convolutional Neural Networks work?" Aug. 2016. [Online]. Available: https://brohrer.github.io/how_convolutional_neural_networks_work.html

[30] M. Seppänen, "GPS-satelliitin radan ennustaminen," Master's thesis, Tampere University of Technology, 2010. [Online]. Available: http://URN.fi/URN:NBN:fi:tty-2011051914680

[31] P. R. Spofford, "The National Geodetic Survey Standard GPS Format SP3." [Online]. Available: ftp://igs.org/pub/data/format/sp3_docu.txt

[32] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for Simplicity: The All Convolutional Net," Dec. 2014.

[33] P. Steigenberger, U. Hugentobler, A. Hauschild, and O. Montenbruck, "Orbit and clock analysis of Compass GEO and IGSO satellites," *Journal of Geodesy*, vol. 87, no. 6, pp. 515–525, Jun. 2013.

[34] N. Takahashi, M. Gygli, B. Pfister, and L. Van Gool, "Deep Convolutional Neural Networks and Data Augmentation for Acoustic Event Detection," in *Interspeech 2016*, Sep. 2016.

[35] J. Vieira, E. Leitinger, M. Sarajlic, X. Li, and F. Tufvesson, "Deep Convolutional Neural Networks for Massive MIMO Fingerprint-Based Positioning," in *International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) 2017*, Aug. 2017.

[36] F. Ye, Y. Yuan, B. Tan, and J. Ou, "A Robust Method to Detect BeiDou Navigation Satellite System Orbit Maneuvering/Anomalies and Its Applications to Precise Orbit Determination," *Sensors (Basel, Switzerland)*, May 2017.

[37] W. Zhang, K. Liu, W. Zhang, Y. Zhang, and J. Gu, "Wi-Fi positioning based on deep learning," in *2014 IEEE International Conference on Information and Automation (ICIA)*, Jul. 2014.