University of West Bohemia
Faculty of Applied Sciences
Department of Computer Science and Engineering

Master's thesis

Automatic detection of sleep spindles

Pilsen 2022 Jan Rychlík

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta aplikovaných věd
Akademický rok: 2021/2022

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Bc. Jan RYCHLÍK**
Osobní číslo:	**A19N0074P**
Studijní program:	**N3902 Inženýrská informatika**
Studijní obor:	**Medicínská informatika**
Téma práce:	**Automatická detekce spánkových vřetének**
Zadávající katedra:	**Katedra informatiky a výpočetní techniky**

Zásady pro vypracování

1. Seznamte se s možnostmi analýzy lidského spánku, prostudujte zejména možnosti analýzy spánkových vřetének.
2. Seznamte se s již dostupným otevřeným datovým souborem vysoce kvalitních spánkových vřetének ohodnocených člověkem.
3. Prostudujte možnosti, algoritmy a použité postupy automatické detekce spánkových vřetének s ohledem na datový soubor z bodu 2.
4. Navrhněte a implementujte vlastní programové řešení pro automatickou detekci spánkových vřetének.
5. Řešení z bodu 4 otestujte na datech z bodu 2 a porovnejte jej s konkurenčními řešení z bodu 3.
6. Zhodnoťte dosažené výsledky.

Rozsah diplomové práce:	**doporuč. 50 s. původního textu**
Rozsah grafických prací:	**dle potřeby**
Forma zpracování diplomové práce:	**tištěná/elektronická**
Jazyk zpracování:	**Angličtina**

Seznam doporučené literatury:

dodá vedoucí diplomové práce

Vedoucí diplomové práce:	**Doc. Ing. Roman Mouček, Ph.D.** Katedra informatiky a výpočetní techniky
Datum zadání diplomové práce:	**10. září 2021**
Termín odevzdání diplomové práce:	**19. května 2022**

L.S.

Doc. Ing. Miloš Železný, Ph.D.
děkan

Doc. Ing. Přemysl Brada, MSc., Ph.D.
vedoucí katedry

V Plzni dne 11. října 2021

Declaration

I hereby declare that this master's thesis is completely my own work and that I used only the cited sources.

Pilsen, 19th May 2022

Jan Rychlík

Acknowledgment

I would like to mainly thank my supervisor Doc. Ing. Roman Mouček Ph.D. for his advice, willingness, and patience. Also, I would like to thank colleagues Ing. Filip Hácha and Ing. Kamil Ekštein Ph.D. for their advice. And Finally, I would to thank Karine Lacource, employee from the MASS data center.

Abstract

Sleep is an integral part of human life, and the average person sleeps about one-third of their life. Therefore, it is important to understand sleep and analyze it correctly. The goal of this master thesis is to propose, design, implement, and test various machine/deep learning methods suitable for EEG signal processing to identify sleep spindles. The learning algorithms were trained on well-annotated data provided by the Montreal Archive of Sleep Studies (MASS) data centre. The best classification result was achieved by the convolutional neuron network with an accuracy of over 67%.

Abstrakt

Spánek je nedílnou součástí lidského života a průměrný člověk prospí asi jeho jednu třetinu. Proto je důležité spánku rozumět a správně ho analyzovat. Cílem diplomové práce je navrhnout, implementovat a otestovat různé typy metod strojového učení vhodné pro zpracování EEG signálu a identifikaci spánkových vřetének. Učící se algoritmy byly natrénovány na anotovaných datech, poskytnutých datovým centrem Montreal Archive of Sleep Studies (MASS). Nejlepšího výsledku klasifikace dosáhla konvoluční neuronová síť s přesností přes 67%.

Contents

1 Introduction

Sleep is an integral part of human life, and the average person sleeps about one-third of his life. During sleep, the human body regenerates. Furthermore, the energy is restored, and the brain is relaxed. Unfortunately, sleeping disorders often occur in today's age, so the research in this area is propitious for the improvements of human sleep.

Brain activities change during sleep. These activities are divided into two main parts: REM and non-REM, then splits into three phases. This master's thesis focuses on finding an artifact in the non-REM phase. This artifact is called sleep spindle.

Brain activities are measurable, and their records are called EEG data. Currently, there are a lot of data but they are not understood nor studied thoroughly. Also, the data contain detailed information which can be easily overlooked by the human eye. Therefore, it is necessary to process the data by computer. The computer eliminates the human mistake factors, always works at 100%, and processes the data faster and more precisely.

One of the options to analyze EEG data with machine learning methods is using neural networks. Neural networks emulates real neurons from the brain. Neurons are interconnected. They transmit signals to each other and transform them using certain transmission functions. To train neural networks used the data described in the article [19].

This master thesis proposes to design, implement and test different types of machine/deep learning methods suitable for EEG signal processing. The research is focused on the use of neural networks such as CNN, LSTM, and Dense and their combinations.

2 EEG data and sleep

The human brain is made up of neurons, and their communication is based on electric impulses. The impulse size is between ten to one hundred microvolts. These impulses measured in time are called Electroencephalography data (EEG data). The brain activity waves are split into sections depending on the frequency and size of the amplitude. The primary waves are called: alfa, beta, gamma, and delta.

2.1 Measurement of EEG data

There is general agreement on the primary source of the EEG signal the EEG arises from synchronized synaptic activity in populations of neurons Excitation of the neurons creates an extracellular voltage near the neural dendrites that is more negative than elsewhere along the neuron. This situation is referred to as a dipole: a region of positive charge separated from a region of negative charge by some distance [9].

There is general agreement on the primary source of the EEG signal EEG arises from synchronized synaptic activity in neuronal populations. Excitation of neurons creates extracellular tension near neural dendrites that is more negative than elsewhere along the neuron. So the electrodes detect the sum of positive and negative charges in their surroundings.

Electrodes read the signal from the head surface; amplifiers bring the microvolt signals into the range where they can be digitalized accurately, converter changes signals from analogue to digital form, and a personal computer (or another relevant device) stores and displays obtained data.

Scalp recordings of neuronal activity in the brain, identified as the EEG, allow measurement of potential changes over time in basic electric circuits conducting between the signal (active) electrode and reference electrode [12]. An extra third electrode, called the ground electrode, is needed for getting differential voltage by subtracting the same voltages showing at active and reference points. The minimal configuration for mono-channel EEG measurement consists of one active electrode, one (or two especially linked together) reference, and one ground electrode. The multi-channel configurations can comprise up to 128 or 256 active electrodes.

For multi-channel montages, electrode caps are preferred, with more electrodes installed on their surface. Commonly used scalp electrodes consist of Ag-AgCl disks, 1 to 3 mm in diameter, with long flexible leads that can be

plugged into an amplifier [7]. AgCl electrodes can accurately record also very slow changes in potential [9]. Needle electrodes are used for long recordings and are invasively inserted under the scalp.

Skin preparation differs; general cleaning of the skin surface from oil and brushing from dried parts is recommended. With disposable and disc electrodes, an abrasive paste is used for slight skin abrasion. With cap systems, an abutting needle at the end of injection is used for skin scraping, which can cause irritation, pain, and infection. Especially when a person's EEG is measured repeatedly, and the cap is mounted for the same electrode points, there is a threat of certain pain and bleeding. That is why the proper hygiene and safety protocol should be kept.

Using the silver-silver chloride electrodes, the space between the electrode and skin should be filled with conductive paste also helping to stick. With the cap systems, there is a small hole to inject conductive jelly. Conductive paste and conductive jelly serve as media to ensure the lowering of contact impedance at the electrode-skin interface.

In 1958, International Federation in Electroencephalography and Clinical Neurophysiology adopted a standardization for electrode placement called the 10-20 electrode placement system [13]. This system standardized the physical placement and designations of electrodes on the scalp. The head is divided into proportional distances from prominent skull landmarks (nasion, preauricular points, inion) to provide adequate coverage of all regions of the brain. Label 10-20 designates proportional to distance in percents between ears and nose where points for electrodes are chosen. Electrode placements are labelled according to adjacent brain areas: F (frontal), C (central), T (temporal), P (posterior), and O (occipital). The letters are accompanied by odd numbers on the left side of the head and even numbers on the right side. The left and the right side are considered by convention from the point of view of a subject.[12]

2.2 Sleep

Introduction

Sleep is an integral part of human life, and the average person sleeps about one-third of his life. During sleep, the human body regenerates. Furthermore, the energy is restored, and the brain is relaxed. Unfortunately, sleeping disorders often occur in today's age, so the research in this area is propitious for the improvements of human sleep.

Brain activities change during sleep. These activities are divided into

two main parts: REM and non-REM, then splits into three phases. This master´s thesis focuses on finding an artifact in the non-REM phase. This artifact is called sleep spindle.

Brain activities are measurable, and their records are called EEG data. Currently, there are a lot of data but they aren´t understood nor studied thoroughly. Also, the data contain detailed information which can be easily overlooked by the human eye. Therefore, it is necessary to process the data by computer. The computer eliminates the human mistake factors, always works at 100%, and processes the data faster and more precisely.

One of the options to analyze EEG data with machine learning methods is using neural networks. Neural networks emulates real neurons from the brain. Neurons are interconnected. They transmit signals to each other and transform them using certain transmission functions. To train neural networks used the data described in the article [19].

This master thesis proposes to design, implement and test different types of machine/deep learning methods suitable for EEG signal processing. The research is focused on the use of neural networks such as CNN, LSTM, and Dense and their combinations.

2.2.1 Sleep Stages

There are two basic types of sleep: rapid eye movement (REM) sleep and non-REM sleep (which has three different stages). Each is linked to specific brain waves and neuronal activity. Sleep cycle through all stages of non-REM and REM sleep several times during a typical night, with increasingly longer, deeper REM periods occurring toward morning.

Stage 1 non-REM sleep is the changeover from wakefulness to sleep. During this short period (lasting several minutes) of relatively light sleep, heartbeat, breathing, and eye movements slow, and muscles relax with occasional twitches. The brain waves begin to slow from their daytime wakefulness patterns.

Stage 2 non-REM sleep is a period of light sleep before the sleep become deeper. Heartbeat and breathing slow, and muscles relax even further. The body temperature drops and eye movements stop. Brain wave activity slows but is marked by brief bursts of electrical activity. Everyone spends more repeated sleep cycles in stage 2 sleep than in other sleep stages.

Stage 3 non-REM sleep is the period of deep sleep that everyone needs to feel refreshed in the morning. It occurs in longer periods during the first half of the night. The heartbeat and breathing slow to their lowest levels during sleep.Muscles are relaxed,and it may be difficult to be awakened. Brain waves become even slower.

REM sleep first occurs about 90 minutes after falling asleep. The eyes move rapidly from side to side behind closed eyelids. Mixed frequency brain wave activity becomes closer to that seen in wakefulness. Breathing becomes faster and irregular, and heart rate and blood pressure increase to near waking levels. Most of-dreaming occurs during REM sleep, although some can also occur in non-REM sleep. Arm and leg muscles become temporarily paralyzed, which prevents from the acting out of dreams. As humans age, the sleep less of their time in REM sleep. Memory consolidation most likely requires both non-REM and REM sleep.[15]

2.2.2 Sleep spindles

Sleep spindles refer to a well recognizable, burst-like sequence of 10–15 Hz sinusoidal cycles in the electroencephalogram (EEG) of sleeping mammals. The name stems from the envelope of a sleep spindle waveform that resembles the shape of the wool-spinning device. It is now 80 yr ago that sleep spindles were first observed in pioneering electroencephalographic studies on naturally sleeping humans.

Since these landmark discoveries, sleep spindles have led to insights into novel organizing principles of mammalian sleep and into how sleep relates to cognitive abilities and disease. Their phasic appearance over a hierarchy of time scales divides non-rapid-eye-movement sleep (NREMS) into time windows with variable cortical states and sensors. Sleep spindles are now becoming a tool to monitor the inner workings of TC loops as they are unconstrained by wakefulness-related activity. These advances increase their diagnostic and therapeutic usefulness and refine approaches to explore their roles in memory consolidation. There are expectations that sleep spindles may ultimately tell us about the development, efficacy, and plasticity of the forebrain circuits that make us intelligent individuals.

Despite this progress, many elementary questions remain open. Whereas we know a lot about the subcortical origins of sleep spindles, what a cortical spindle is in terms of its neuronal activity profile across cortical layers is just beginning to be addressed. Although the methodology of sleep spindle detection is in steady improvement, where and when they appear on the

cortical surface once generated in thalamic circuits is not yet clear.

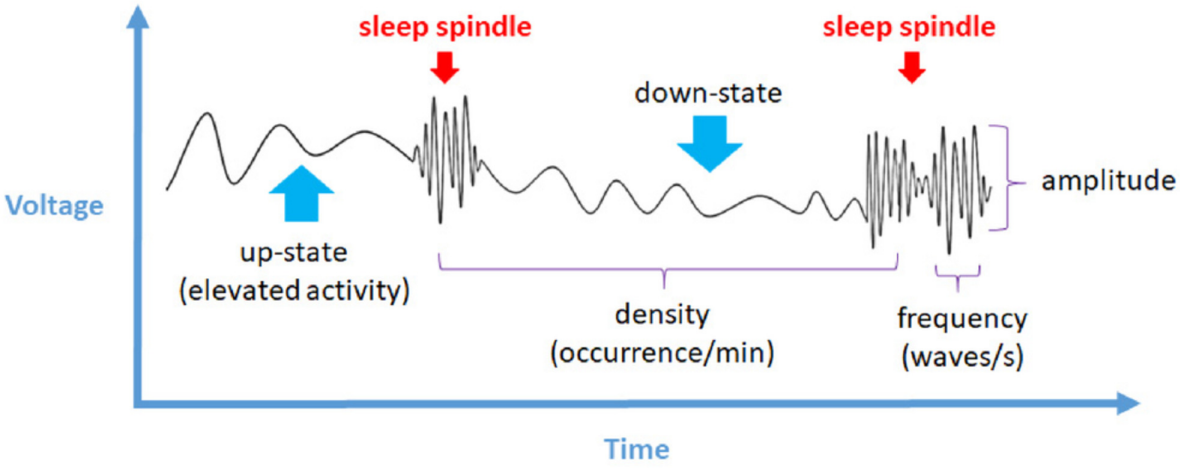

Figure 2.1: Example of EEG sleep spindles. [8]

The figure 4.1 shows voltage magnitude in time with two sleeping spindles. It manifests itself in higher amplitude and higher wave density (frequency). The sleep spindles can be preceded by two states: up-state and down-state. The states determine the position compared to the previous spindle.

3 State of the Art

This chapter deals with state of the art in the field and provides information about similar researches at this area.

3.1 Massive online data annotation.

A Canadian group of researchers compared and evaluated the performance of three subtype scorers: "experts, researchers, and non-experts", as well as seven spindle detection algorithms. Their findings show that only two algorithms had performance scores similar to human experts. Furthermore, the human scorers agreed on the average spindle characteristics (density, duration, and amplitude), but there were significant age and sex differences (also observed in the set of detected spindles). This study demonstrates how the MODA (Massive Online Data Annotation) platform can be used to generate a highly valid open source standardized dataset for researchers to train, validate and compare automated detectors of biological signals such as the EEG. [10]

3.1.1 Dataset

The Montreal Archive of Sleep Studies (MASS) is an open-access and collaborative database of laboratory-based polysomnography (PSG). Its goal is to provide a standard and easily accessible source of data for benchmarking the various systems developed to help the automation of sleep analysis. It also provides a readily available source of data for fast validation of experimental results and exploratory analyses. Finally, it is a shared resource that can be used to foster large-scale collaborations in a sleep study. [19]

3.1.2 Methods

Researchers tested the success of annotating sleep spindles by individual groups of people. The first group represents the experts on sleep scoring. The second group was composed of people without any experience, but they had just a short online training. As part of the research, they compared the results of expert, non-expert and non-learning methods

In band-pass filters (11–15 Hz) the EEG signal is used to compute its envelope. An upper (8 x mean) and lower (2 × means) thresholds are used to detect spindles.
Band-pass filters (11.3–15.7 Hz) the EEG signal is used to compute the root mean squared (RMS) on sliding windows (100 ms length with a step of 50 ms), and then applies a threshold (1.5 × STD).
Band-pass filters (11–15 Hz) the EEG signal is used to compute the RMS on sliding windows (25 ms length with a step of 25 ms), and then the filter applies a threshold (95th percentile).
The method transforms the EEG signal into continuous Morlet wavelets to compute the moving average on sliding windows (0.1 sec length), and then applies the threshold (4.5 × mean).
The method computes the absolute (Mean Square) sigma (11–16 Hz) power, the relative sigma power with Power Spectral Analysis, the covariance and correlation between sigma filtered and the unfiltered EEG signal on sliding windows (0.3 sec length with a step of 0.1 sec). It then detects a spindle if the four features extracted from EEG exceed their respective threshold (1.25 µV2, 1.6 × STD, 1.3 × STD and 69%).
The method decomposes the EEG signal into three components: Direct Current, oscillation around 13.5 Hz and other frequency components (0.3–30 Hz). Spindles are detected from the oscillation around 13.5 Hz with an upper (2.33 × STD) and lower (0.1 × STD) thresholds applied in sliding windows (60-sec length).
The method decomposes the EEG signal into three components: transient (t), low-frequency (lf) and oscillations (s). five are represented sparsely with Short Time Fourier Transform (1.28 sec length with a step of 0.32 sec). It then detects spindles by thresholding (c1 = 0.03) the Teager-Kaiser energy operator (energy smooth) of s band-pass filtered (11.5–15.5 Hz). Parameters initialization: lambda0 = 0.6, lambda1 = 7, lambda2 = 8.5, mu = 0.5 and c1 = 0.03.

Table 3.1: The table with short description of methods used in the research [10].

3.1.3 Results

Polysomnographic data from 180 subjects were sourced from the Montreal Archive of Sleep Studies (MASS). The dataset was split into two "parts", where part 1 consisted of 100 younger subjects (mean age of 24.1 years) and part 2 consisted of 80 older subjects (mean age of 62.0 years). A subset of N2 stage sleep from the C3 channel was sampled from each subject (see table 3.1.2 methods for details). 25-sec epochs of this single-channel EEG

were presented to expert PSG technologists, researchers, and non-expert scorers via a custom web-based scoring platform. Users identified the start and stop of candidate spindles, and indicated their confidence (high, med, low) for each spindle marked. In total, 47 PSG technologists, 18 researchers, and 695 non-experts viewed 10,453, 6,636, and 37,467 epochs respectively in Phase 1. Phase 2 was viewed by 31 PSG technologists (7,941 epochs viewed). No scorers viewed the whole dataset and the histogram of the number of scorer views per epoch. A minimum number of scorers per epoch was crucial to compile a reliable gold standard (GS): the median number of scorers per epoch is 5 for the PSG technologists, 4 for researchers, and 18 for non-experts. More than 95% of all the epochs have been seen by at least 3 PSG technologists. Almost 100,000 candidate spindles were identified by all scorers combined.

3.2 A deep learning algorithm based on 1D CNN-LSTM for automatic sleep staging

Technol Health Care [25] used the LSTM and CNN neural networks to annotate three phases of human sleep (Wake, REM, Non-REM) on Sleep-EDF dataset. The achieved accuracy was 93.47% using the Fpz-Cz electroencephalogram channel.

3.2.1 Dataset

The sleep-EDF database is an open-source dataset and contains 197 whole-night PolySomnoGraphic sleep recordings, containing EEG, EOG, chin EMG, and event markers. Some records also contain the values of respiration and body temperature. Corresponding hypnograms (sleep patterns) were manually scored by well-trained technicians according to the Rechtschaffen and Kales manual, and are also available.

3.2.2 Methods

Long Short-Term Memory (LSTM), convolutional neural networks (CNNs), and their combination were used for Sleep-EDF dataset processing. LSTM is a time recurrent neural network. LSTM can process and predict important events with long intervals and delays in time series. LSTM includes the input gates, the output gates, and the forget gates.

CNN is frequently used to recognize two-dimensional images but it was found successful also for EEG patterns processing. It combines the local perception, weight sharing, and down sampling. Local perception allows hidden units not to be full connections. Weight sharing enables different signals to share a convolution kernel.

The proposed CNN-LSTM algorithm contains seven layers to realize automatic sleep staging. Four layers of 1D CNN and three layers of LSTM were used.

3.2.3 Results

In this study, a thirteen layers model was proposed for the classification of sleep stages. Performance comparisons of different classification algorithms based on single-channel EEG, single-channel EOG. A deep learning algorithm based on 1D CNN-LSTM for automatic sleep staging a combination of EEG and EOG signals was presented. In addition, each group finished five test sets to make the results more convincing. The results showed that the model also had good classification performance for different physiological signals. When the EEG signal of the Fpz-Cz channel was used, the accuracy was 93.47%. When the EEG signal of the Fpz-Cz channel and the EOG signal were used, the accuracy of the staging reaches 94.15%. The algorithm had high accuracy and generalization ability. It could achieve more accurate sleep automatic staging without manual extraction features. In the future, this model can be used in sleep-related research. The fully automatic system could replace the traditional error-prone large-scale PSG signal manual expert inspection task. At the same time, deep learning itself is also an algorithm worthy of continuous research, especially for distinguishing sleep periods with high similarity.

3.3 Automated scoring of pre-REM sleep in mice with deep learning

FH Achen University of Applied Sciences, in cooperation with the department of Medical Engineering and Technomathematics in Germany, uses a machine learning method to detect three phases of mice sleep, Wake, REM, and Non-REM. They used a simple neural network for scoring.

3.3.1 Dataset

The dataset consists of polysomnographic recordings of 18 mice with a total recording duration of 52 days. Each mouse was recorded for three days. The data were acquired during a study at the University of Tübingen which tested the influence of dietary variations on sleep. All mice (age about ten weeks, male) were kept at the animal facility at the University of Tübingen, Germany. The mice were chronically implanted with a synchronous recording device of two EMG and four EEG electrodes seven days before the first recording. Four stainless steel screws were placed on top of the cortex to record the EEG signal, and two flexible stainless steel wires were inserted into the neck muscles to measure the EMG signal [7].

The parietal right EEG time series were low-pass filtered using a 4th order Butterworth low-pass filter (critical frequency: 25.6 Hz). Then the set of training data was balanced. At the end of the data preparation synthetic data were added for the more extensive training set [7].

3.3.2 Methods

The network architecture, which draws upon experience gained in previous work by some of the authors, consists of a feature extractor and a classifier. Before entering the feature extractor, the time series is batch normalized ("Batchnorm", see Fig. 3.1). The feature extractor consists of eight convolutional layers, each of which is composed of 96 kernels of size d×1×5d×1×5, where d denotes the depth of the output volume of the previous layer. Since one time series enters the network, d=1 for the first convolutional layer, while d=96 for all other convolutional layers. To downsample the time series information along with the feature extractor, every other convolutional layer uses a stride of 2, with the other layers having a stride of 1. The resulting convolved signals of each layer are nonlinearly transformed by rectified linear units (ReLUs) and subsequently batch normalized (Batchnorm). Furthermore, authors apply Dropout regularization to every other convolutional layer. Finally, the output (also called features) of the feature extractor is concatenated into a vector ("Flatten", see Fig. 3.1), and Dropout regularization is applied before these features enter the classifier [7].

The classifier is composed of two fully connected (FC) layers with 113*96 = 10848 and 80 neurons, respectively. The first FC layer is equipped with rectified linear units (ReLUs) as nonlinearity and Dropout regularization, while the second FC layer uses a softmax activation function to determine the score probabilities p→ , which represents the activations of the neurons of the output layer. The output layer consisted of three output neurons when

the network was trained to distinguish between major sleep stages (Wake, REM, NREM). When the network was trained to also score pre-REM sleep and segments containing artifacts, the output layer consisted of five neurons [7]. The layer structure is shown in figure: 3.1.

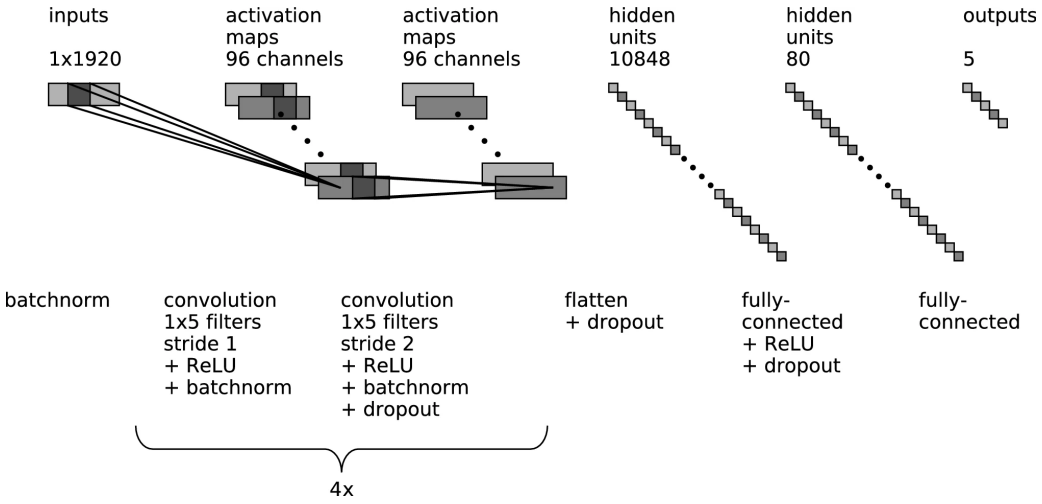

Figure 3.1: Network architecture. The feature extractor consists of eight convolutional layers, and the output of the last convolutional layer is flattened (Flatten). EEG time series data is batch normalized (Batchnorm) before the first convolutional layer is applied. The classifier consists of two fully connected layers. A softmax layer translates the activations of the previous layer into class probabilities. [7]

3.3.3 Results

The results obtained for the network trained without data augmentation show that it scored more than 98% of the Wake segments, more than 94% of the REM segments, and close to 92% of the NREM segments were classified correctly. The network was also able to score 71% of the artifact segments correctly. The pre-REM stage was scored in 58% of the segments correctly.

The results also show that more than 96% of the segments in the test set predicted as Wake, more than 87% of the segments predicted as REM, and 97% of the segments predicted as NREM had been assigned the predicted stage by the human expert. Furthermore, more than 50% of the segments that were predicted as an artifact by the network had been assigned artifact as the true stage. Predictions of pre-REM had pre-REM as the true stage in nearly 41% of cases, while in about 50% of cases, predictions had NREM and about 9% had REM as the true stage.

3.4 EEG signal classification using LSTM and improved neural network algorithms

The Soft Computing compares classification methods to classify preictal, postictal, and interictal classes in EEG. They have chosen SVM, logistic regression machine learning algorithms and NN for EEG signal classification. The model is composed of two-layer LSTM and four-layer improved NN deep learning algorithms. The novelty lies in one-dimensional gradient descent activation functions with radial basis operations used in the initial layers of improved NN, which help achieve better performance. In addition, statistical features, namely mean, standard deviation, kurtosis, and skewness are extracted for input EEG [13].

3.4.1 Dataset

The EEG record comes from the Bonn database measured in 2018. The dataset was cleaned, filtered, and normalized as part of other research.

The training set with annotation was created by analysts from soft computing(non-experts).

3.4.2 Methods

SVM, logistic regression, and NN neural network were chosen as conventional methods.

SVM

The key of the SVM classifier is to select a proper kernel function. There exists many classical kernel methods, namely perceptron, linear, RBF and polynomial. The regulation, kernel parameters (C, y) and their optimal combination give a low general testing error (GTE). SVM deals better even with large number of elements and small training data. SVM takes the output structure and spatial contiguity as features and can give higher classification accuracy. Boosting, AdaBoost and Wang's boost algorithms for SVM learn the imbalanced data better and give good results [13].

Performance of SVM can be improved by using changes in time complexity as O(n3). However, it neglects the classification of observation positions and position similarities. New setbacks can be introduced by deciding the distance-based weights and sign-based classifier to get an improved ensemble classifier. SVM acts as a base classifier for imbalanced data learning

and analysis. It is the classifier which uses kernel function efficiently. Modified boosted and Wang boosted SVMs are two types of SVM for further improvement in the performance of SVM [13].

Logistic regression

The Laplacian method for selecting features and sparse logistic regression gives better performance compared to univariate analysis in this case. Euler elastic-based multi-nominal logistic regression can be applied for multi-class classification and can work with minimum energy and overcome the overfitting problem. Total variation logistic regression (TVLR), Euler elastic logistic regression(EELR) and sparse logistic regression (SLR) are various methods that can be applied to EEG for classification. Among these three, EELR works better in terms of finding a larger number of active regions and discriminative regions for brain using fMRI as input in analysis [13].

NN neural network classification

A neural network can work based on different wavelets; hence, the neural network can combine feature extraction, and classification steps together. A wavelet-based neural network acts as a binary classifier, and hence, the combination and decision strategy to be used are based on the N-class problem in different applications.

A multi-dimensional feedforward neural network is used for regression analysis with multi-variables. R represents a radial function. Input vectors are accepted and then applied to the radial function followed by the one-dimensional wavelet function. A linear combiner is used at the end to combine all the signals for classification. Computational time adjusting the parameters and weights in the path are the challenging task in neural network architectures [13].

LSTM classification

Long short-term memory (LSTM) is based on a recurrent neural network (RNN), which is a deep learning algorithm. RNN consists of recurrent structures which locally feed the firing strength; thereby, external registers or memories are not required to store previous outputs. Computational complexity is low in LSTM due to recurrent structures used in RNN [13].

Improved neural network classification

To improve the performance of the neural network, an improved neural network with four layers is designed with relu, relu, relu and sigmoid activation functions in each layer, respectively. It also uses the Adam optimizer and binary cross-entropy loss model. In general, there is no fixed condition on the number of neurons to be selected in the hidden layer.The Fully connected layer is used throughout the neural network [13].

3.4.3 Results

The worst of these five algorithms is logistic regression. This method gets an accuracy of about 55%. The fourth one is the NN neural network, with an accuracy of about 59%. Finally, in the middle is the SVM algorithm with an accuracy of 6%. Just ahead is LSTM, with an accuracy rate of 68%, and the best one is improved NN, with an accuracy of over 76%.

The improved NN network layer contains four layers, with NADAM optimizer, logcosh loss function, and activation functions on layers are relu, relu, relu, and sigmoid.

3.5 Conclusion of state of the art

The first mentioned research, Massive online data annotation, crowdsourcing to generate high-quality sleep spindle annotations from EEG data, describes several non-learning methods to annotate sleep spindles and get accuracy between 55% and 71%. However, it is also shown that experts annotated with an accuracy of over 82% and non-experts get over 78%. The score from experts and non-expert is similar. The results between the expert group and the non-expert were very supportive. Because training a non-expert took little time, this success led to the idea of neural networks. In the second research, a deep learning algorithm based on 1D CNN-LSTM for automatic sleep staging. The researchers tested CNN, LSTM, and their combination to annotate all sleep phases: REM, nonREM, and wake. The accuracy was about 93% at the Fpz-Cz channel.

Automated scoring of pre-REM sleep in mice with deep learning uses the neural network to annotate sleep (REM, nonREM, wake) phases of mice sleep. The researchers used dense neural networks for the classification. The neural network detected individual artifacts in the EEG records with an accuracy of over 72%.

The research called: "EEG signal classification using LSTM and improved neural network algorithms" classify EEG signal in two ways. The study tested regression and SVM from the group machine learning methods. However, deep learning methods used NN, LSTM, and improved NN networks.

4 Neural networks

An artificial neural network is one of the computational models used in artificial intelligence. Its model is the behaviour of the corresponding biological structures. An artificial neural network is a structure designed for data processing. Each network consists of an input layer, an output layer, and 0-n hidden layers. Each layer is composed of artificial neurons that are interconnected and pass values to each other. The values are then processed by a function inside the neuron and the output value is also one of the inputs in the next layer of the neural network.

4.1 Neuron

A biological neuron is made up of the body of a neuron. Inside the body is a core to processing information. There are also protrusions on the neuron. Short protrusions are called dendrites and are used to receive impulses into a cell. The long-ones are called axons and are used to pass information out of the cell. Figure 4.1 shows the similarities between the neurons.

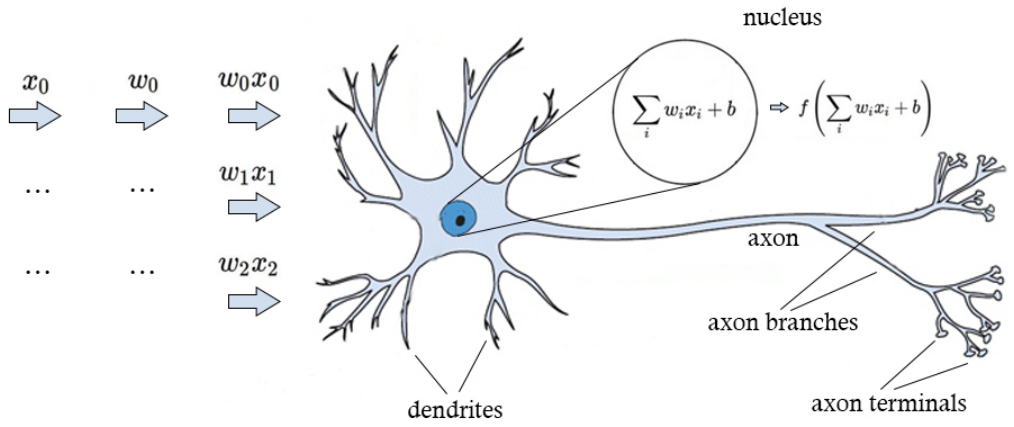

Figure 4.1: Comparison of a biological and artificial neuron. [2]

Figure 4.3 shows the basic unit of the artificial neuron network - the neuron.

- X_i is the neuron input signal value, output of the previous neuron.

- w_i A synapse is multiplied with a corresponding special value known as a parameter or weight. The parameter is determined during the learning phase from your training set. In a neuron with m synapses, there are m parameters. In other words, every synapse has a corresponding parameter. It should be noted that parameters can occur within intervals such as [-1, 1] or [0, 1] and, for the first feed forward they were setup randomly. [20]

- $\sum$ A Linear combiner of input values.

$$\sum_{1}^{n} X_i * w_i$$

- **b** Bias does not allow local minima to rest when learning a neural network, but it still slightly considers that the algorithm will diverge to a global minimum. Every neuron, except the first layer, has a bias weight input.

$$(\sum_{1}^{n} X_i * w_i) + b$$

- **f** A special function known as the activation function limits the amplitude of the output of the neuron. In other words, it normalizes the output to an interval such as [0, 1] or [-1, 1]. [20]

$$f((\sum_{1}^{n} X_i * w_i) + b)$$

- **y** - Output of the neuron.

4.2 Layer

The layer in the neural network is a group of neurons, and each layer performs the same function. The neurons calculate the weighted sum of inputs and execute the activation function.

4.2.1 Input layer

The input layer servers for reading the init values from external sources. Then performs the calculations via neuron activation functions and passes the values to each neuron in the next hidden layer.

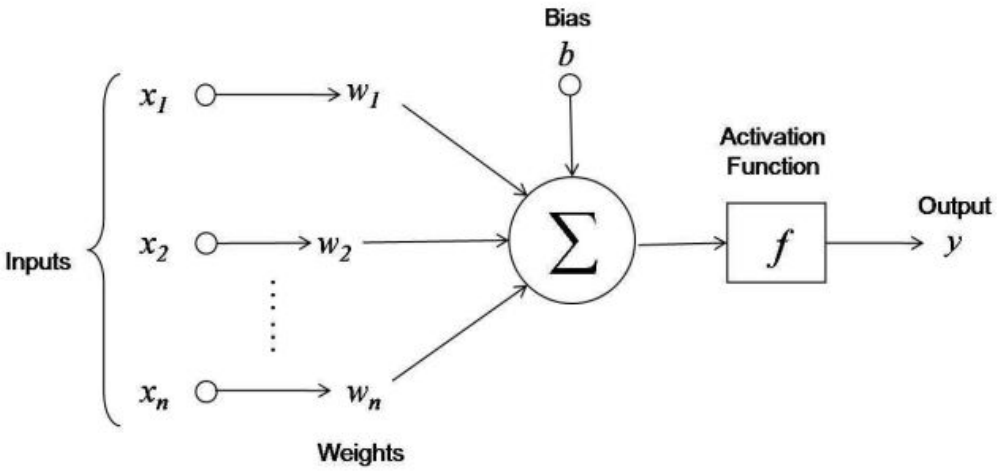

Figure 4.2: Artificial neuron. [22]

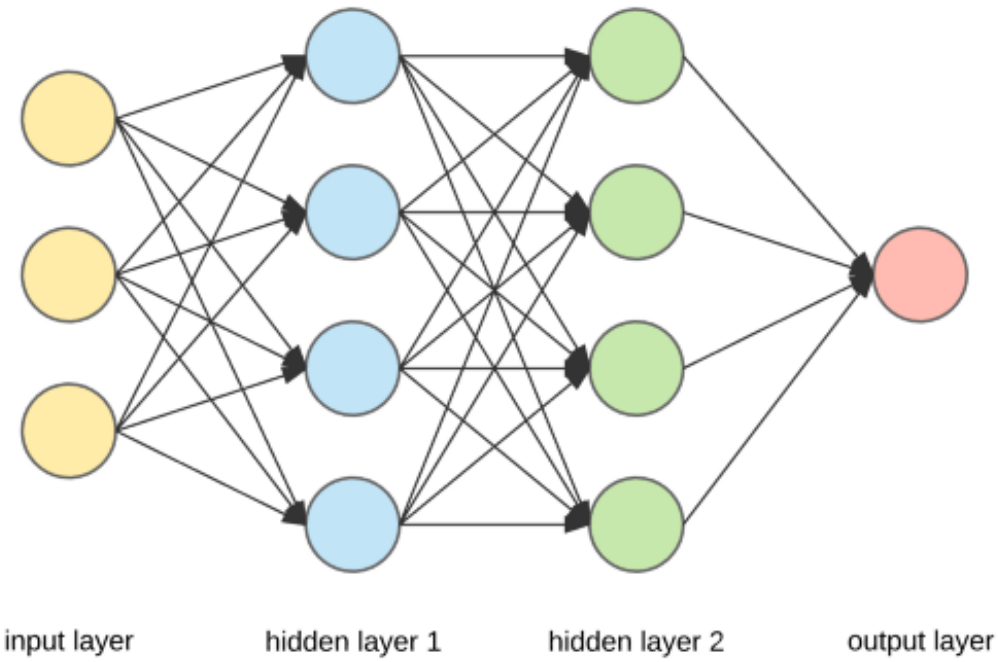

Figure 4.3: Example of standard neural network. The network has one input layer, two hidden layers, and one output layer. [16]

4.2.2 Hidden layer

The hidden layers are between the input and output layers. In the neural network could be zero or more hidden layers. The hidden layers make the networks superior to most of the machine learning algorithms. The neurons

in each layer calculate the weighted inputs, add the bias, and provide the value as the input value for each neuron in the next layer.

4.2.3 Output layer

The output layer is responsible for the results of neural networks. The neuron with the highest value is considered as the result.

4.3 Backpropagation

The goal of backpropagation is to compute the partial derivatives

$$\frac{\partial C}{\partial w} \; and \; \frac{\partial C}{\partial b}$$

of the cost function C with respect to any weight w or bias b in the network. For backpropagation to work we need to make two main assumptions about the form of the cost function.

$$C = \frac{1}{2n} \sum ||y(x) - a^L(x)||^2$$

Where: n is the total number of training examples; the *sum* is over individual training examples: $x; y = y(x) y = y(x)$ is the corresponding desired output; L denotes the number of layers in the network; and $aL = aL(x) aL = aL(x)$ is the vector of activations output from the network when x is input.

The first assumption we need is that the cost function can be written as an average:

$$C = \frac{1}{n} \sum_x C_x$$

over cost functions Cx for individual training examples, x. This is the case for the quadratic cost function, where the cost for a single training example is $C_x = \frac{1}{2}||y - a^L||^2$. This assumption will also hold true for all the other cost functions.

The assumption is that what backpropagation actually lets us do is compute the partial derivatives $\frac{\partial C_x}{\partial w}$ and $\frac{\partial C_x}{\partial b}$ for a single training example. We then recover $\frac{\partial C_x}{\partial w}$ and $\frac{\partial C_x}{\partial b}$ by averaging over training examples. In fact, with this assumption in mind, we'll suppose the training example x has been fixed, and drop the x subscript, writing the cost Cx as C. We'll eventually put the x back in, but for now, it's a notational nuisance that is better left implicit [14].

4.4 Dense neural network

In any neural network, a dense layer is a layer that is deeply connected with its preceding layer which means the neurons of the layer are connected to every neuron of its preceding layer. This layer is the most commonly used in artificial neural network networks [24]. The dense layer's neuron in a model receives output from every neuron of its preceding layer, where neurons of the dense layer perform matrix-vector multiplication. Matrix vector multiplication is a procedure where the row vector of the output from the preceding layers is equal to the column vector of the dense layer. The general rule of matrix-vector multiplication is that the row vector must have as many columns as the column vector [24].

4.5 LSTM neural network

The LSTM(Long Short-Term Memory) neural network comes from the recurrent neural network. The RNNs are networks that work on actual inputs while considering prior outputs. Because the RNN networks have problems with long-term dependencies and exploding and disappearing gradients that occur during the backtracking training phase of a network are another concern with RNNs. The last problem related to no finer control over which aspects of the context should be preserved and how much should be forgotten [23].

The main distinction between RNN and LSTM designs is that the LSTM's hidden cell is a gated unit or gated cell. It is made up of four layers that interact with one another to generate the cell's output as well as the cell's state. After then, these two items are passed on to the next concealed layer. LSTMs contain three logistic sigmoid gates and one tanh layer [1], unlike RNNs, which have only one neural net layer of tanh [23]. Gates were developed to limit the amount of data that could travel through the cell. They figure out which parts of the data will be needed by the following cell and which will be discarded. The result is generally in the 0–1 range, with 0 indicating "reject all" and 1 indicating "include all [23].

LSTM recurrent unit and its work

- **Hidden state & new inputs** — a hidden state from a previous timestep (h_t-1) and the input at a current timestep (x_t) are combined before passing copies of it through various gates [4].

[1] Tanh layer is a hidden layer with Tanh activation function

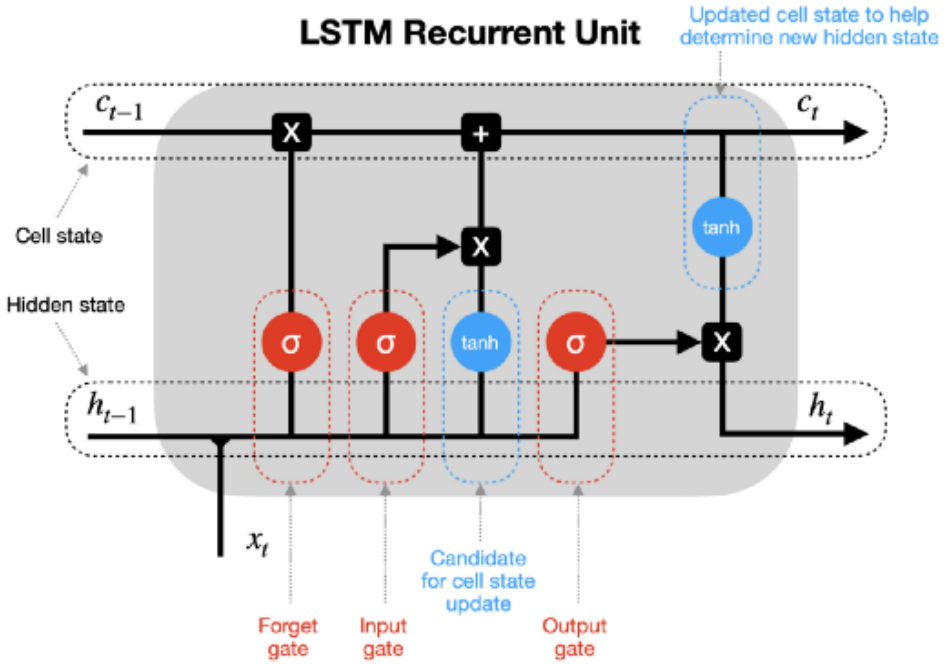

Figure 4.4: LSTM neural network scheme. [4]

- **Forget gate** — this gate controls what information should be forgotten. Since the sigmoid function ranges between 0 and 1, it sets which values in the cell state should be discarded (multiplied by 0), remembered (multiplied by 1), or partially remembered (multiplied by some value between 0 and 1) [4].
- **Input gate helps** to identify important elements that need to be added to the cell state. Note that the results of the input gate get multiplied by the cell state candidate, with only the information deemed important by the input gate being added to the cell state [4].
- **Update cell state** —first, the previous cell state (c_t-1) gets multiplied by the results of the forget gate. Then we add new information from [input gate × cell state candidate] to get the latest cell state (c_t) [4].
- **Update hidden state** — the last part is to update the hidden state. The latest cell state (c_t) is passed through the tanh activation function and multiplied by the results of the output gate [4].
- Finally, the latest cell state (c_t) and the hidden state (h_t) go back into the recurrent unit, and the process repeats at timestep t+1. The

loop continues until we reach the end of the sequence.

4.6 Convolutional neural network

Convolutional neural networks (CNN) are more often utilized for classification and computer vision tasks. Prior to CNNs, manual, time-consuming feature extraction methods were used to identify objects in images. However, convolutional neural networks now provide a more scalable approach to image classification and object recognition tasks, leveraging principles from linear algebra, specifically matrix multiplication, to identify patterns within an image. That said, they can be computationally demanding, requiring graphical processing units to train models [5].

The convolutional network structure

The convolutional network hidden structure is consisted of these layers:

- Convolutional Layer
- Pooling Layer
- Fully-Connected Layer

The convolutional layer is the core building block of a CNN, where the majority of computation occurs. It requires a few components: input data, a filter, and a feature map. Let us assume that the input will be a colour image, which is made up of a matrix of pixels in 3D. This means that the input will have three dimensions — height, width, and depth which corresponding to RGB in an image. We also have a feature detector, also known as a kernel or a filter, which will move across the receptive fields of the image, checking if the feature is present. This process is known as a convolution [5].

The feature detector is a two-dimensional (2-D) array of weights, which represents part of the image. While they can vary in size, the filter size is typically a 3x3 matrix; this also determines the size of the receptive field. The filter is then applied to an area of the image, and a dot product is calculated between the input pixels and the filter. This dot product is then fed into an output array. Afterwards, the filter shifts by a stride, repeating the process until the kernel has swept across the entire image. The final output from the series of dot products from the input and the filter is known as a feature map, activation map, or a convolved feature [5].

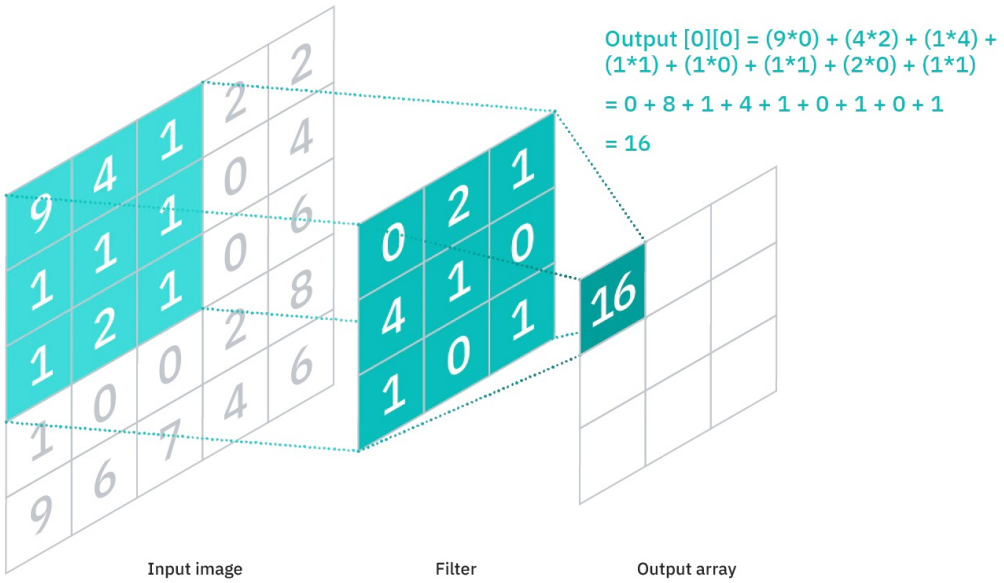

Figure 4.5: Application of filter on an image inside a CNN network. [5]

As we can see in the figure 4.5, each output value in the feature map does not have to connect to each pixel value in the input image. It only needs to connect to the receptive field, where the filter is applied. Since the output array does not need to map directly to each input value, convolutional (and pooling) layers are commonly referred to as "partially connected" layers. However, this characteristic can also be described as local connectivity [5].

The pooling Layer, also known as downsampling, conducts dimensionality reduction, reducing the number of parameters in the input. Similar to the convolutional layer, the pooling operation sweeps a filter across the entire input, but the difference is that this filter does not have any weights. Instead, the kernel applies an aggregation function to the values within the receptive field, populating the output array [5].

The Full-connected layer appropriate describes itself. As mentioned earlier, the pixel values of the input image are not directly connected to the output layer in partially connected layers. However, in the fully-connected layer, each node in the output layer connects directly to a node in the previous layer.

This layer performs the task of classification based on the features extracted through the previous layers and their different filters. While convolutional and pooling layers tend to use ReLu functions, FC layers usually leverage a softmax activation function to classify inputs appropriately, producing a probability from 0 to 1.

5 Analysis

Reasonable analysis requires a thorough check of the data, their characteristics, and possible solutions. Because this task is unique, there is no single resolution but only several methods that lead to a proper solution. The whole analysis is based on the knowledge from chapter 3; State of the Art and experience of a neural network or medicine experts.

The first idea leads to using a neuron network with three layers and thirty-two or sixty-four neurons in the input layer and one neuron in the output layer. Unfortunately, this input layer size is shown as small and inadequate during the testing. For this reason, an input layer with 512 neurons started to be used during the research. Such a large input layer can accommodate the entire sleep spindle at the entrance.

After discussing the master thesis topic with the specialist on human sleep, there are specified channels where the sleep spindle is the most significant.

A further specialist in neural networks recommended expanding the neural network to seven layers. Also, the expert suggested using another Python library due to use in his projects.

The last layer is extended by one neuron. CUrrently are two neurons in the output layer. The first neuron represents the spindle, and the second represents the non-spindle. The signal part is classified into a class represented by a neuron with a higher value.

5.1 Data

This section describes the data used for the master's thesis. It contains information about the data source,and its description, and preprocessing.

5.1.1 Data source

The Montreal Archive of Sleep Studies (MASS) in Canada was created as a part of the project: "Massive online data annotation, crowdsourcing to generate high-quality sleep spindle annotations from EEG data" [10] extensive database of EEG signals. Each EEG record was annotated by experts on sleep and sleep parts.

MASS

The Montreal Archive of Sleep Studies (MASS) is an open-access and collaborative database of laboratory-based polysomnography (PSG) recordings. Its goal is to provide a standard and easily accessible source of data for benchmarking the various systems developed to help the automation of sleep analysis. It also provides a readily available source of data for fast validation of experimental results and for exploratory analyses. Finally, it is a shared resource that can be used to foster large-scale collaborations in sleep study. [19]

MASS is composed of cohorts comprising subsets which split records by specific properties. Recordings within subsets are kept as homogeneous as possible, whereas they are more heterogeneous between subsets. [19]

Currently, the first MASS cohort available is described in [18]. This cohort comprises polysomnograms of 200 complete nights recorded in 97 men and 103 women of age varying between 18 and 76 years (mean: 38.3 years, SD: 18.9 years).[19]

MASS database access

The following documents conditions were created to access the MASS database.

- **Research description**
- **Research affiliation**
- (Optional) **Request for the approval of ethics committee**
- **Approval of ethics committee**
- (Optional) **Code for work with the data** - The MASS institute requires any certificate or training in the area: Data and work with them. Because of these training are not open source. There was created an agreement with the MASS data-center on this code.
- **Mass license** - The MASS institute assigns the license document after delivering all documents mentioned above.

It took more than three months to meet these conditions and communicate with all institutions.

5.1.2 Data description

Data from MASS are divided into several cohorts. Each cohort represents a specific project focused on data collection, the data from the cohort C1 are chosen for the research. The C1 cohort is folded into five stages (SS) and each stage represents a group of EEG measures with specific properties. The properties of all sets are described in table 5.1.2 below.

		SS1	SS2	SS3	SS4	SS5
The number of recordings	Male	34	8	28	14	13
	Female	19	11	34	26	13
	Total	53	19	62	40	26
Average age	Male	63.5	24.3	40.4	27.4	23.0
	Female	63.7	23.2	44.2	24.1	26.9
	Total	63.3	23.6	42.5	25.3	25.0
Annotations	Sleep spindles	NA	Experts	NA	NA	NA
	Muscular artefacts	Automatic	NA	Automatic	NA	Automatic
	Apne-hypopnea	Experts	NA	NA	NA	NA
	Micro arousal	Experts	NA	NA	NA	NA
	PLMS	Experts	NA	NA	NA	NA

Table 5.1: The table contains information on the number of measurements in cohort C1 and their properties.

Stages contain three types of files. First, in the data folder are *Base.edf* and *PSG.edf* files. The Base.edf files contain metadata about measurements as time data, age, sex of the subject, impedance of electrodes at the start of the action, the impedance at the end of action, etc. The measurements processed, filtered, and clean data are saved in the PSG files. The third file is located in the annotations folder and contains all annotations about measurements. The sleep spindles are annotated in stage SS2. The sleep spindle is characterized by two parameters. The first parameter is the timestamp when the spindle starts, and the second parameter is the duration in seconds.

Data from cohort C1 and phase SS2 were selected as the training data set. This phase was chosen because it is the only one that contains an annotation of sleep spindles.

5.1.3 Data preparation

One measurement contains about eight hours of EEG recordings. The record includes all recurrent sleep phases(REM, non-REM, and their sub-phases). Anyway, the sleep spindles appear in the specific parts of sleep, and their occurrence is unique. Sleep spindles occur on average in two percent of the record. Therefore, it was necessary to use set balancing. Another problem is the size of the measured values in the signal. The measured values range is between 10^{-4} and 10^{-6} Volts. Therefore, normalization was necessary.

Spindles set balancing

To process the signal through the neural network, the EEG records are divided into smaller parts and each piece of signal is splitted into sets using annotations. Unfortunately, the group without spindles contains about ninety-eight percent of all pieces of the signal. This unbalanced set makes it impossible to learn on neural networks. Also, removing parts of the signal from the collection by a specific pattern distorts the learning because the neural network can predict the pattern to balance sets. For balancing data set it is possible to use a random number generator.

Technical information about the data

Cohort 1 (C1)	
Memory size	35.2 GB
Measurement frequency	250Hz
Total number of measurements	200
Average length of one EEG measure	7H 59min

Table 5.2: Information about cohort C1.

The table 5.1.3 represents the information about data in stage SS2.

EDF

The data are provided in the EDF format. The EDF format (stands for European Data Format) was designed to store medical time series data; it is most commonly used format for EEG data. A non-breaking extension was added in 2003 called EDF+. It is an open, documented standard that is agnostic to any recording system or hardware/software supplier. The initial document describing the format was published in 1992 [3].

Stage 2 (SS2)	
Memory size	7.26 GB
Total number of measurements	19
Total number of spindles	11204
Average number of spindles in measure	217
Time length	151H
Maximum number of samples on spindle	569
Minimum number of samples on spindle	86
Maximum duration of spindle	2.218605 sec
Minimum duration of spindle	0.335915 sec
Average length of one EEG measure	7H 59min

Table 5.3: Information about stage SS2.

The original EDF files are particularly easy to read; they consist of two header blocks followed by all the data. The first header block contains various information about the recording, the device specification, ADC range and filters. Pertaining to the data, it contains the number of data records, the length of one record and a number of "signals". All of the fields in the two headers are plain human readable ASCII characters. The header is 256 bytes long [3].

5.2 Methods

In the chapter State of the art 3, there are introduced methods for working with EEG signals. In the mentioned research, in which neural networks were used, the LSTM or Convolutional models with good results were used. However, these types of neural networks were used in many other types of research. Some methods were not mentioned in the chapter State of the Art but could work on EEG data. One of them is base attention. Therefore, these methods are the most appropriate for this research. The following is a list of methods that can be used to process an EEG signal.

5.2.1 Dense neural network

The dense neural network is a primary, most straightforward neural network in the Keras TensorFlow library. The results from the dense implementation are used to be compared with results from other implementations.

5.2.2 LSTM neural networks

The LSTM neural networks were create to work with signals, using knowledge from the previous inputs. Because the sleep spindles appear in one phase of sleeping, the option of knowledge with previous input can be helpful.

5.2.3 CNN neural networks

Also, the convolutional networks are mentioned in a lot of research. They are initially designed for processing matrices (two dimensional array - pictures). For analysis of EEG signals is necessary to modify the CNN network to 1D.

5.2.4 CNN - LSTM neural networks

The combination of these types of neural networks is not common. However, this combination is used in research: A deep learning algorithm based on 1D CNN-LSTM for automatic sleep staging [25], and achieved excellent results. However, in the case of sleep spindle annotating is very difficult to set up the layers correctly. Therefore, draft for the neural network was taken from the research mentioned in the chapter 3.

5.2.5 Value-based method

The last method is not from the deep learning group. The principle of this method is that it sums the measured values and compares the resulting values with each other. For this method, it is crucial to set the border correctly. Everything below the border is marked as a non-spindle, and everything above the border is marked as a spindle. This method was never used in other research but was chosen on the basis of data analysis.

5.2.6 Transformer

A transformer model is a neural network that learns context and thus meaning by tracking relationships in sequential data like the words in a sentence [11].

Transformer models apply an evolving set of mathematical techniques, called attention or self-attention, to detect subtle ways even distant data elements in a series influence and depend on each other [11].

5.3 EEG signal

The sleep spindle is marked in the yellow rectangle in the figure 5.1. The density of the peaks is in some channels higher. Also, the size of the peaks is larger than in out of the sleep spindle.

These properties are more significant only in some channels. Therefore, after discussing with the doctor and expert on sleeping at University Hospital in Pilsen, the following channels were selected: EEG Cz-CLE EEG C3-CLE, EEG C4-CLE, EEG P3-CLE, EEG P4-CLE, and EEG Fpz-CLE.

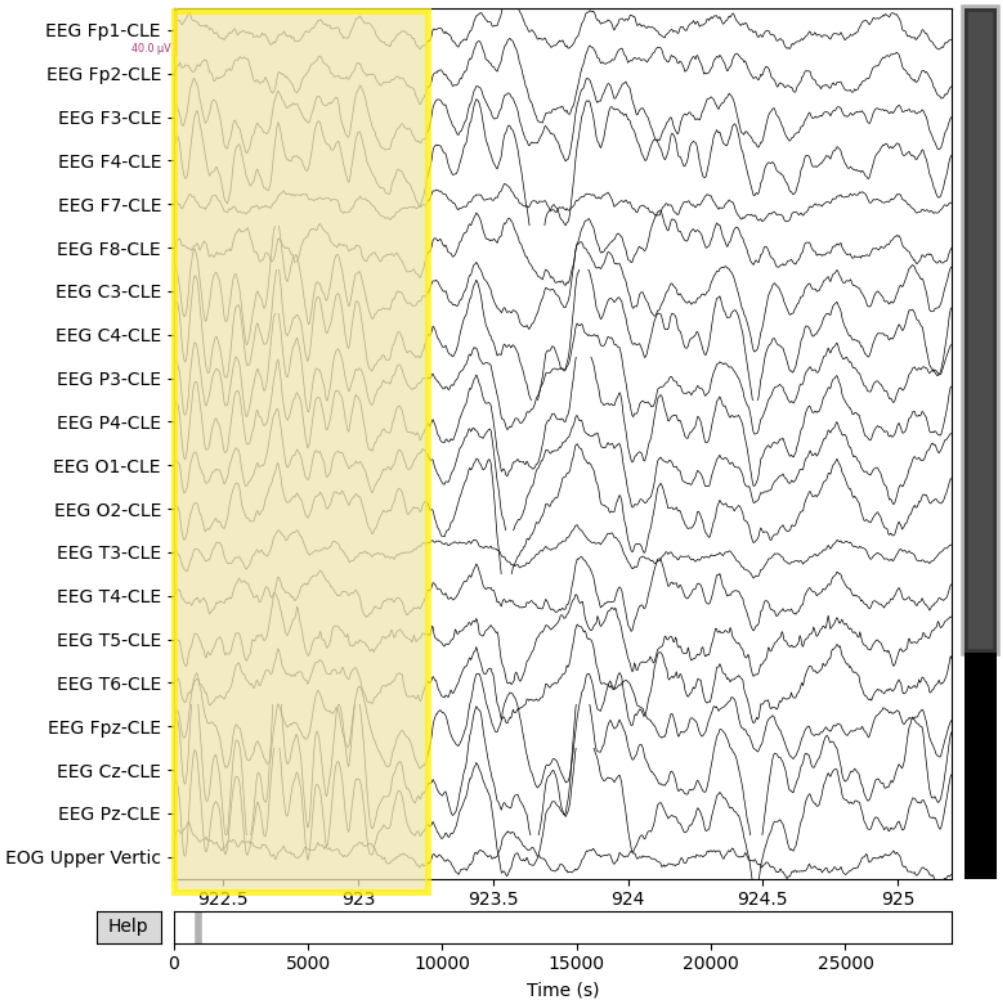

Figure 5.1: Example of sleep spindles displayed across all signals (In the yellow rectangle).

6 Software Requirement Specification

The goal of this chapter to provide requirement specification of a software system to be developed.

6.1 Overall Description

6.1.1 Product Perspective

The application's primary goal is to prepare data from the MASS data set and come with deep learning classification algorithms suitable for such data. Algorithms need to be designed, implement, test, evaluate, and choose the most aqueous processing.

6.1.2 Product Features

The research serves to test LSTM, CNN, Dense, and their combinations to score sleep spindles.

6.1.3 User Classes

Researcher

Researches can use the software as an inspiration for the following investigations. Also, the results can be used as a starting position for the subsequent scoring algorithms.

Doctors

In case of excellent results, the application can be used by doctors to process the EEG recordings of individual patients.

6.1.4 Operating Environment

The program has to run on a standard computer on Windows and Linux platforms. The repository selected for storing the code and learned neural network is the Git-Hub platform. The Git-Hub meets all-important features.

Its a open open source and available for everyone and it is a common platform in the programmer's community.

6.1.5 Design and Implementation Constraints

There are a couple of conditions for the application. The first one is the implementation in Python. The python language is most widespread among research centers.

The second condition is that all created documentation has to be in English or French requested by the MASS data center.

6.1.6 Documents

Three documents need to be created during this research. The first one is a diploma thesis. The second documents are created to request the data. The last document is a user manual as a part of diploma thesis.

6.1.7 Assumptions and Dependencies

The first dependency is to gain access to the MASS database. The access to the database is conditioned by six documents. Everything about getting the permit is described in section MASS 5.1.1.

7 Architecture and Design

This chapter describes the overall architecture and design of the software tool for scoring sleep spindles, including dependencies and used libraries, except for standard python libraries, neural network libraries, and graphical libraries.

7.1 Programming Language

Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and is not specialized for any specific problems. This versatility, along with its beginner-friendliness, have made it one of the most-used programming languages today. A survey conducted by industry analyst firm RedMonk found that it was the second-most popular programming language among developers in 2021 [17].

7.2 Packages and Dependencies

This section describes the used packages in the whole research work.

MNE

MNE package is a tool for exporting, analyzing, and visualizing human electrography data focusing on neurophysiological data. The package is used in the *PyData.py* script for reading data in *.edf* format.

PyEDFlib

This package is also used in the *PyData.py* for reading data as the second option.

Numpy

NumPy is a python library created for working with multidimensional arrays. This package is used in all scripts. However, NumPy is a fundamental pillar for the libraries with neural networks.

Pandas

Pandas is a package for data analysis. Among other things, it is also used in neural networks libraries.

Matplotlib

This package serves for a graphical representation of the results of neural networks.

Torch

Torch is the library for neural networks. This library is used in *PyConvolutional1DTorch.py*.

Cuda

Cuda is a GPU driver supported by torch. This driver is used to speed up the calculation.

Tensorflow

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML-powered applications.

TensorFlow is originally developed by researchers and engineers working on the Google Brain team within Google's Machine Intelligence Research organization to conduct machine learning and deep neural networks research. The system is general enough to be applicable in a wide variety of other domains, as well [6].

Keras

Keras is a more simple API implemented over TensorFlow libraries. The advantage of Keras is that minimizes the number of user actions required for common use cases.

8 Implementation

The implementation of the research work is split into three phases(data preparation, neural network, other machine learning method). The First of all, for data preparation is implemented the script *PyData.py*. In the second phase implements scripts for neural networks. In the last step is implemented script for machine learning (Value-based method).

8.1 Data

List of scripts used to prepare data:

- PyData.py

The data phase contains only one script. The script serves to read data from files by uses package MNE, counts statistics about data, and creates datasets for the learning methods. The main task of the script is pass through the data and process them according to the selected options.

While reading the data, the data visualizer can be turned on to plot the spindles in the chart.

From the statistic part, the following information about data are counted:

- The number of spindles
- The number of of non-spindles
- Min/max length of spindles
- Max number of tics[1] of sleep spindles
- An average number of tics of sleep spindles

Data preparation for the neural networks can be divided into two basic methods (Gradually access, Directly access). These methods differ in accessing data and are described further.

[1] A tic is a one value in EEG measurement

8.2 Gradually access

In the first case, each EEG record is scanned in the specified interval (by default, the size of the neural network's input layer), and each part of the signal is processed. This method gets the part of the EEG signal of the interval size and marks it as a spindle or non-spindle during data storage. In this case, it is possible to define the size of the interval. It is also possible to set up the method to divide the intervals into several classes: spindle, the beginning of the spindle, non-spindle, and end of the spindle.

8.3 Directly access

The second method gets the sleep spindle directly. This method uses knowledge about sleep spindle start, length, and maximum size of sleep spindle over all datasets. Moreover, this method guarantees that it always takes the whole spindle.

The data balancing method

The aim of this method is to randomly reduce the set with non-spindle records. A number on the interval <0.99> is generated before the non-spindle is saved. If this number equals 50 (probability is 1%), the spindle values are stored in the field. This field is then saved to a file.

Listing 8.1: "The example of data balancing method"

```
1 if (random.randint(0, 100) == 50):
2     dataX.append(layerInput)
3     dataY.append(noSpindle) #[1,0,0,0]
```

File structure

The prepared data are split into two files. In the first file represents EEG signal values, and the second file contains information on whether the line with EEG values represents the sleep spindle. Files that contain information on whether the values are a sleep spindle exist in three variants.

- **variant A** The variant A expect one neuron in the input layer. The value 1 represents the spindle and value 0 represents non-spindle.
- **variant B** The variant B is designed for two output layer. In each line are two values.

- **variant C** The C variant is created for four neurons in the input layer. Each line represents four classes: non-spindle, start-spindle,spindle ,and end-spindle

Listing 8.2: Example of file with EEG signal values "

```
1 −1.6219501e−05;−1.619539e−05;...;−1.3434882e−05
2 −1.4121995e−05;−1.19039e−05;...;−1.2627222e−05
3 −1.3025024e−05;−1.470061e−05;...;−9.420691e−06
4 −5.4065e−06;−3.85145e−06;...;−5.4788e−06
5 .
6 .
7 .
```

Listing 8.3: Example of file with annotations; A variant

```
1 0.0      //Non−spindle
2 1.0      //Spindle
3 1.0
4 0.0
5 .
6 .
7 .
8 .
```

Listing 8.4: Example of file with annotations; B variant

```
1 [0.0,1.0]      //Non−spindle
2 [1.0,0.0]      //Spindle
3 .
4 .
5 .
```

Listing 8.5: Example of file with annotations; C variant

```
1 [1.0,0.0,0.0,0.0]      //Non−spindle
2 [0.0,1.0,0.0,0.0]      //Start−spindle
3 [0.0,0.0,1.0,0.0]      //Spindle
4 [0.0,0.0,0.0,1.0]      //End−spindle
5 .
6 .
7 .
```

Training dataset

The stage SS2 contains 19 measured EEG recordings. The first five-teen recordings are used as a training dataset, and the last four are used to verify neural networks. During the preparation was counted over 15 thousand samples for training the and 4.5 thousand records for validation were counted.

8.4 Neural network

The scripts for working with neural network are the following ones:

- PyConvolution1DTorch.py
- PyConvolutional1D.py
- PyDense.py
- PyLSTM.py

First, the values of EEG signal are normalized at intervals <1,-1>. The xMin and xMax are values from the set for neural network training. For data to be validated, normalization is also performed.

Listing 8.6: "Method to normalize EEL values"

```
1    xMin = trainX.min()
2    xMax = trainX.max()
3    trainX = (trainX-xMin) / (0.5*(xMax-xMin))-1.0
```

Then the data enter the neural networks [2]. All the neural networks had implemented in rudiment the ADAM optimizer and binary-crossentropy loss function.

8.4.1 PyConvolution1DTorch.py

The only script is implemented in the Torch package. Currently, the neural network is folded from seven layers. The first four layers are CNN with the Relu activation function. Then, three dense layers are followed with softmax as the activation function. Finally, the output layer contains two neurons. Also, there are implemented methods for displaying the learning rate, cost function, etc.

[2]The current states of implementation of neural networks is not the one that achieved the best results, but it is description of current state.

8.4.2 PyConvolutional1D.py

This script is based on the Keras-TensorFlow libraries. There are two CNN layers and two dense layers with one neuron at the output layer with the sigmoid as the activation function.

8.4.3 PyDense.py

The basic neural network model composed from three dense layers. The sixty-four neurons are in the input layer and one neuron at the output layer. All the neurons have sigmoid as the output function.

8.4.4 PyLSTM.py

The last script from the part of the neural networks is implemented to test the LSTM networks. There are four layers. The first one is the embedding layer. Two hidden layers are LSTM. The output layer is dense with one neuron with a sigmoid activation function.

8.5 Machine learning

The following scripts are implemented for machine learning methods:

- PyBruteForce4Channels.py
- PyBruteForce.py

The scripts read data from the file and sum all values of the EEG signal in the file line. This counted value is compared with the border. The border starts at zero value and is incremented by value 0.1. Everything up the edge is classified as the sleep spindle.

Listing 8.7: Method for evaluating the result

```
1  border = 0
2  same =0
3  diff = 0
4  while (1):
5      spindle = 0
6      if(valueInLine > border):
7          spindle = 1
8
9      if(spinlde == annotation[toTestedLine]):
10         same++
11     else:
12         diff++
13     border = border + 0.1
```

The engine of the method is shown in the example. The valueInLine represents the sum of the values of one line. If the line value is higher than the border, the line is classified as the spindle. The result of the method is a number of correctly classified lines.

The correct value of the border is known when the ratio between correctly identified and misidentified results starts to decrease again.

9 Results

This chapter summarizes the results of all implemented methods. First of all, are introduced the prepared dataset. Then are described the results of neural networks, and the last are showed the results from other learning methods. All neural sites are compared by use an accuracy metrics. All attempts were run on the following computer; its parameters are available in table 9.1

Information about PC	
Processor	Intel(R) Core(TM)i7-8565U CPU @1.80GHz 1.99 GHz
RAM	16GB
System type	x64
Operating system	Windows 11 Home
Producer	ASUS
GPU	NVIDIA GeForce MX150
Driver version	511.69

Table 9.1: The parameters of PC used for all experiments.

9.1 Data

The generated datasets for all methods are shown in Table 9.2. Each dataset has a specific number, and this number is used in tables with the results of neural networks. Therefore, the datasets contain balanced sets of spindles and non-spindles. The "access methods" are described in the chapter 8. Table column: "Input neurons" provides the size of the input layer. The number in "Classes" column represents the number of neurons in the output layer.

The choice of individual datasets for neural network training was realized according to the suitability of the dataset (e.g. the number of neurons in the input or output layer). Also, the neural networks were trained on others datasets, but without success.

Epoch

An epoch means training the neural network with all the training data for one cycle. In an epoch, we use all of the data exactly once. A forward pass

Access method	Input neurons	Classes	Intended network	Description	Dataset No.
Gradual	64	1	All		1.
Gradual	64	2	All		2.
Gradual	32	1	All		3.
Gradual	32	2	All		4.
Gradual	64	4	All		5.
Gradual	64	2	CNN, LSTM	4 Channels	6.
Direct	569	1	CNN		7.
Direct	569	2	CNN		8.

Table 9.2: Prepared datasets for using learning algorithms.

and a backward pass together are counted as one pass [1].

9.2 Dense neural network

The Dense neural network is a primary, most straightforward neural network from the Keras, TensorFlow package. The results from this part of the research will be comared with results of others neural network modes. The testing model is composed of three layers. The complete results of Dense neural network [1] are in Table 9.3. The results of the last three experiments

Dataset No	Input Neurons	Output Neurons	Optimizer	Loss Function	Processing time	Accuracy	Epochs
2.	64	2	SGD	B-C	306 sec	52.2%	10
1.	64	1	SGD	B-C	320 sec	62.3%	5
1.	64	1	SGD	B-C	623 sec	62.3%	10
3.	32	1	ADAM	B-C	75 sec	65.04%	1
3.	32	1	ADAM	B-C	537 sec	65.04%	10
3.	32	1	ADAM	B-C	4834 sec	65.04%	100

Table 9.3: The results of classification using a dense neural network.

show that the best results are obtained on dataset no. 3. It also turned out that it does not depend on the number of epochs performed. All computations are preformed on the CPU.

9.3 LSTM

The LSTM is composed of five layers. The first layer is Embedded. Two layers inside are LSTM with 150 neurons. The last hidden layer is a dense layer with 200 neurons, and the number of neurons in the output layer is specified in Table 9.4 showing also the classification results.

Because balancing randomly chooses the non-spindles intervals, this property can worst the results. All the results are computed at the CPU.

9.4 CNN-Keras

A convolutional neural network implemented in Keras has five layers. The first three are convolutional with 64 neurons in each layer. The last hidden layer is dense with 100 neurons. The results of the neural network are shown in the table 9.5.

[1] The dense model was used as the first one in the research, and the first output from this network had an accuracy of about 98%. This was caused by the poor relationship between the number of non-spindles and spindles. The ratio of these sets was also 98%.

Dataset No	Input Neurons	Output Neurons	Optimizer	Loss Function	Time to process	Accuracy	Epoch
2.	64	2	ADAM	B-C	33 sec	52.81%	1
2.	64	2	ADAM	B-C	250 sec	52.81%	5
2.	64	2	ADAM	B-C	2832 sec	52.81%	100
2.	64	2	ADAM	B-C	11835 sec	52.81%	500
5.	64	4	ADAM	B-C	40 sec	36.19%	1
5.	64	4	ADAM	B-C	227 sec	36.19%	10
5.	64	4	ADAM	B-C	10398 sec	36.19%	500

Table 9.4: The classification results using LSTM neural network.

Dataset No	Input Neurons	Output Neurons	Time to process	Accuracy	Epoch
1.	64	1	183 sec	58.63%	100
1.	64	1	867 sec	58.63%	500
7.	64	1	162 sec	60.78%	100
7.	64	1	759 sec	60.87%	500
8.	64	1	205 sec	59.96%	100
8	64	1	1096 sec	59.96%	500

Table 9.5: The classification using CNN-Keras neural network.

The convolutional neural networks implemented in the Keras package get an accuracy of about 59% This accuracy sort this model to the middle of the results of all tested models.

9.5 LSTM-CNN

A combination of CNN and LSTM networks is implemented in the Keras package. The first one is CNN with 64 neurons. The hidden layer is LSTM with 128 neurons, and the output layer is described in Table 9.6.

Dataset No	Input Neurons	Output Neurons	Time to process	Accuracy	Epoch
1.	64	1	10 sec	52.81%	100
2.	64	1	598 sec	52.81%	100
6.	64	1	2923 sec	52.81%	100

Table 9.6: The classification using LSTM-CNN neural network.

Of course, more variants were experimented here, but all results followed ratio between the number spindles and the non-spindles.

9.6 Convolutional neural networks implemented in Torch

The convolutional part of the neural network is implemented in the Torch package. This network contains seven layers. The first four layers are convolutional ones with the 32 neurons in one layer. A linear layer with 384 neurons follows this. The penultimate layer is also linear and contains 64 neurons. Finally, the last layer contains two neurons. The complet structure is shown on the figure 9.1.

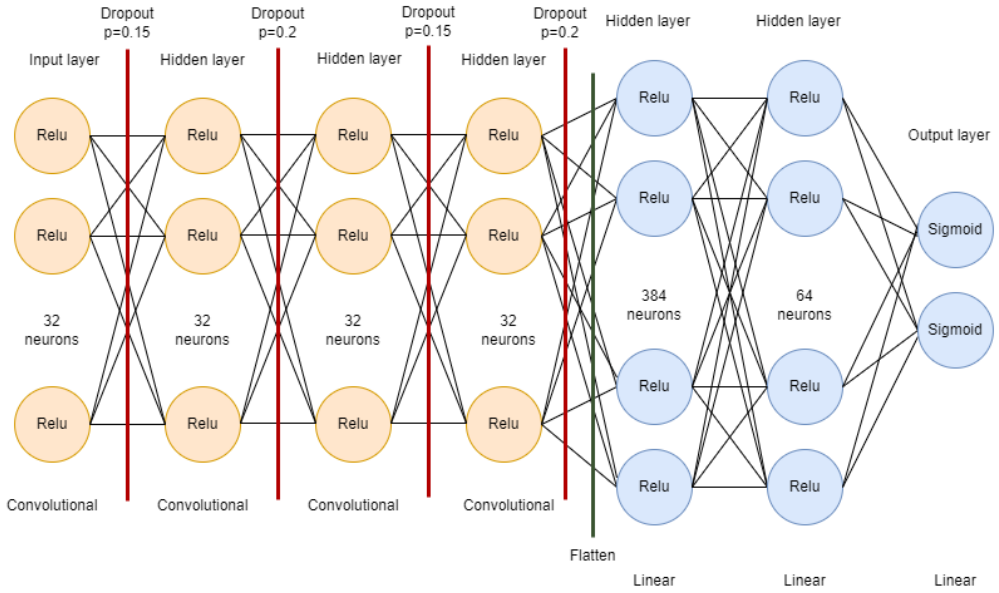

Figure 9.1: The CNN neural network net. Dropout layer prevents stuck in the local minimum; Flatten is the link between CNN and Linear layer.

The results were computed on the GPU. Table 9.7 shows that this result [2] is the best of the performed measurements. The CNN neural network gets an accuracy of a little over 67% and needs 2000 epochs to train. The table 9.8 describes the neural network.

The graphs show the accuracy value 9.2 and the loss function 9.3 on the validation dataset for the best option of neural network. In the first graph 9.2 the value from the accuracy is on the y-axis and in the second graph 9.3

[2]The same network implemented with the Keras package does not achieve such results.

Dataset No	Input Neurons	Output Neurons	Time to process	Accuracy	Epoch
1.	64	1	10 sec	53.27%	1
1.	64	1	598 sec	57.73%	100
1.	64	1	2923 sec	60.73%	500
1.	64	1	7835 sec	60.73%	1000
2.	64	2	732 sec	57.29%	500
2.	64	2	1021 sec	63.89%	1000
2.	64	2	1455 sec	65.96%	1500
2.	64	2	2836 sec	67.15%	2000
2.	64	2	3637 sec	67.15%	2500

Table 9.7: the classification using CNN-torch neural network.

	CNN Layers				Linear layers		
Number of neurons	32	32	32	32	384	64	2
Activation function	Relu	Relu	Relu	Relu	Relu	Relu	Softmax

Table 9.8: The structure of neural networks with the best classification results.

the output value from the loss function is on the y-axis. Both graphs have the number of attempts at the validation set on the x-axis.

9.7 Value-Based Method

The goal is to find the optimal border, which divides the sums in line of measured values into spindle and non-spindle categories.

Dataset No	Time to process	Accuracy	Optimal Border
1.	1265 sec	58.63%	10.20000007
6.	5289 sec	64.12%	48.79999999
7.	4706 sec	63.99%	39.57999999

Table 9.9: The results of the the Value-based method.

This simple method achieved quite good results. In total, it annotated 64% of the input. However, the result is similar to neural networks, but there are a lot of possibilities with an enormous potential to improve the result. These possibilities were discussed in the chapter 10.

the figure 9.4 shows the case when the border iterates to the best state.

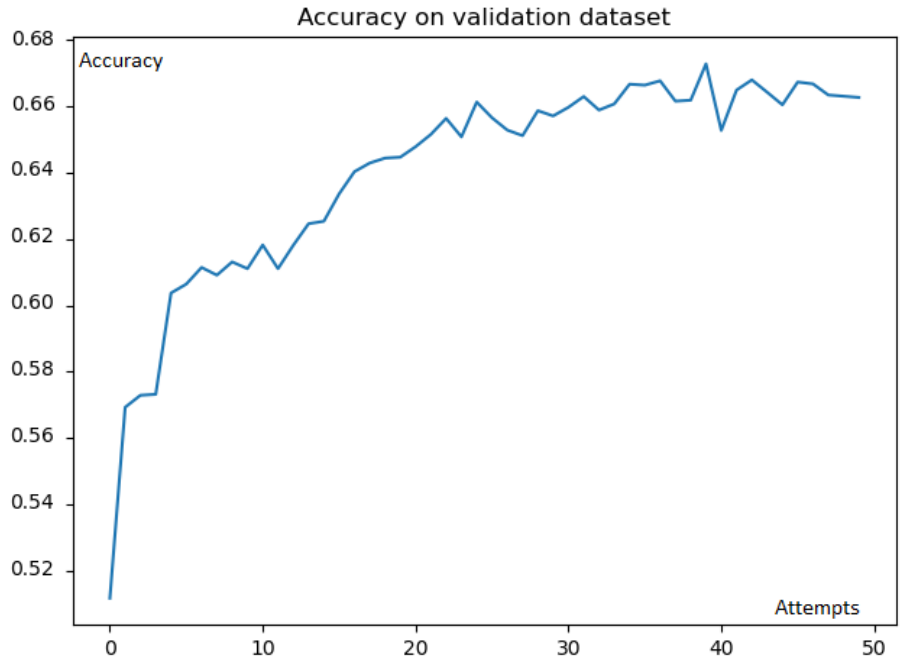

Figure 9.2: The accuracy on validation dataset; X-axis = number of attempts at the validation set; Y-axis = the accuracy value.

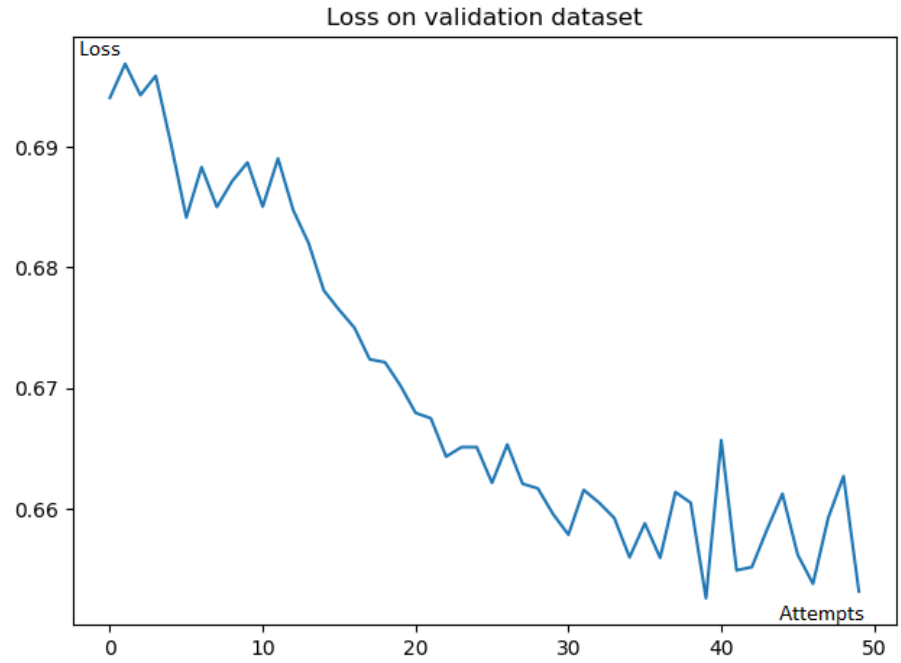

Figure 9.3: The loss function on validation dataset; X-axis = number of attempts at the validation set; Y-axis = the loss value.

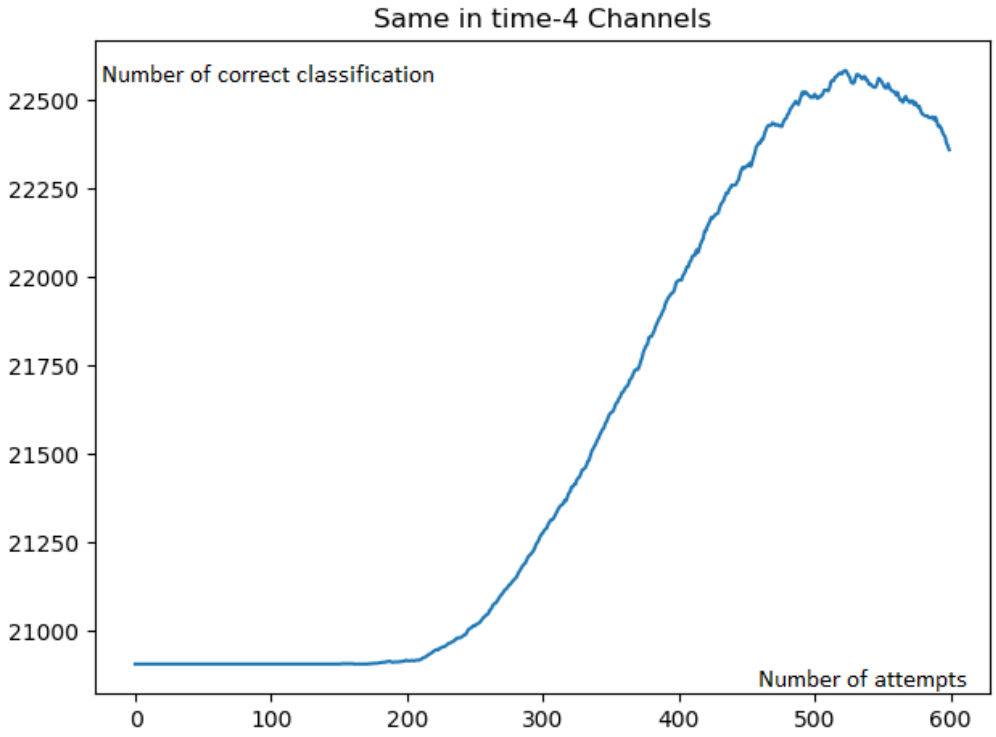

Figure 9.4: Graph: axe X = Number of attempts, axe Y = Number of same results

10 Conclusion

The master's thesis analyzes and processes the EEG signal measured during sleeping state and searches for sleep spindles inside using neural networks. For the thesis it was necessary to request data, which was provided by the MASS data center from Canada. These data were used because they are clean,filtered, and well-annotated.

The data were first balanced and split into training and testing datasets.Then they were used to learn neural networks. The research was tested and implemented in more than five ways.

From the group of neural networks was tested Dense, Convolutional, LSTM, and their combinations. The worst result got the LSTM and a variety of LSTM-CNN neural networks. These networks got an accuracy of 52.81%. This result was the only ratio between spindles and non-spindles sets. Slightly better results got the Dense and CNN networks from the Keras packages. Their accuracy rate was around 60-65%.

The best result has the CNN neural network implemented in the Torch package. The neural network had four CNN layers and three linear layers. The complex network is described in Table 9.8. This network had accuracy with over 67%.

The classification results achieved in this thesis are comparable with the classification results provided in the papers presented in the State of the art chapter. The best results to classify artefacts in the EEG in the State of the were 70-75% accuracy.

The valued-based method got a classification accuracy of over 63%. These results were a little less then in State of the Art.

The figure 10.1 shows the individual classification methods and their best accuracy achieved during the research.

10.1 Discussion

When using neural networks, there are a large number of possibilities that can cause massive number of parameters change. The experiments with the Torch implementations have shown that CNN networks were promising to solution.

The results from value-based method lead to the use a hybrid neural network. The hybrid neural network was tested in the Keras package, with

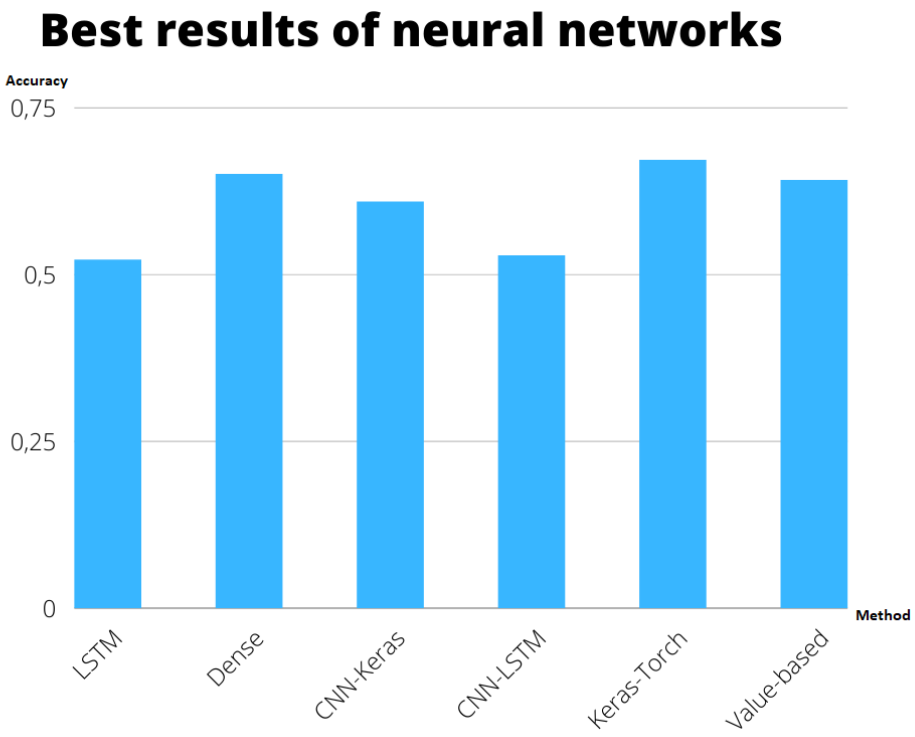

Figure 10.1: The best classification resutts of neural networks achieved during the thesis.

an outcome accuracy of over 55%. So the next step should be to implement this variant in Torch libraries.

The next option is to implement an attention-based model. Since these models are suitable for speech recognition, they may be suitable for calibrating sleep spindles.

The results of the value-based methods show that if the whole spindle or more channel is used, the success of the technique increases by six percent. More channels added makes it possible to predict that using the entire spindle over all channels can increase the success rate again.

10.2 Author's word

The thesis goals were fulfilled. Unfortunately, there has not been enough time to try many other promising methods. The estimated time was two years of research to claim whether the research was on the right way.

All scripts were uploaded and available on the author's git-hub [21], but the repetition needs a new request for the data in the MASS center.

List of Figures

List of Tables

Bibliography

[1] Baeldung. *Epoch in Neural Networks* [online]. Baeldung, 2021. [cit. 2022/05/06]. Epoch in Neural Networks. Available at: https://www.baeldung.com/cs/epoch-neural-networks.

[2] Bolton, K. *A quick introduction to neural networks* [online]. Kris Bolton, 2018. [cit. 2022/01/06]. Git Hub. Available at: https://krisbolton.com/a-quick-introduction-to-artificial-neural-networks-part-1.

[3] Bourke, P. *EDF and EDF+ file format* [online]. Paul Bourke, 2020. [cit. 2022/05/03]. Available at: http://paulbourke.net/dataformats/edf/.

[4] Dobilas, S. *LSTM Recurrent Neural Networks — How to Teach a Network to Remember the Past* [online]. Saul Dobilas, 2021. [cit. 2022/05/03]. Available at: https://towardsdatascience.com/lstm-recurrent-neural-networks-how-to-teach-a-network-to-remember-the-past-55e54c

[5] Education, I. C. *Convolutional Neural Networks* [online]. IBM Cloud Education, 2020. [cit. 2022/05/03]. Available at: https://www.ibm.com/cloud/learn/convolutional-neural-networks.

[6] Flow, T. *Tensor Flow documentation* [online]. Tensor Flow, 2022. [cit. 2022/05/03]. Available at: https://github.com/tensorflow/tensorflow.

[7] Grieger, N. – J.T.C., S. – al., W. S. Automated scoring of pre-REM sleep in mice with deep learning. *Automated scoring of pre-REM sleep in mice with deep learning.* 05 2021, 11. Available at: https://doi.org/10.1038/s41598-021-91286-0.

[8] Iotchev, I. B. – Kubinyi, E. *Shared and unique features of mammalian sleep spindles – insights from new and old animal modelss* [online]. Ivaylo Borislavov Iotchev,Eniko Kubinyi, 2021. [cit. 2022/01/04]. Available at: https://onlinelibrary.wiley.com/doi/full/10.1111/brv.12688.

[9] Jackson, A. F. – Bolger, D. J. The neurophysiological bases of EEG and EEG measurement: A review for the rest of us. *The neurophysiological bases of EEG and EEG measurement: A review for the rest of us.* 7 2014, 51. Available at: https://doi.org/10.1111/psyp.12283.

[10] K, L. – B, Y. – Mednick. Massive online data annotation, crowdsourcing to generate high quality sleep spindle annotations from EEG data. *Massive*

online data annotation, crowdsourcing to generate high quality sleep spindle annotations from EEG data. 10 2020, 7. Available at: https://doi.org/10.1038/s41597-020-0533-4.

[11] MERRITT, R. *What Is a Transformer Model?* [online]. NVIDIA, 2022. [cit. 2022/01/06]. Transformer model. Available at: https://blogs.nvidia.com/blog/2022/03/25/what-is-a-transformer-model/.

[12] Michal, T. Fundamental of EEG Measurement. *MEASUREMENT SCIENCE REVIEW*. 01 2002, 2. Available at: https://www.researchgate.net/publication/228599963_Fundamental_of_EEG_Measurement.

[13] Nagabushanam, P. – George, T. – Radha, S. . EEG signal classification using LSTM and improved neural network algorithms. *EEG signal classification using LSTM and improved neural network algorithms.* 11 2019, 7, s. 24. Available at: https://doi.org/10.1007/s00500-019-04515-0.

[14] Nielsen, M. *How the backpropagation algorithm works* [online]. Michael Nielsen, 2019. [cit. 2022/01/04]. Available at: http://neuralnetworksanddeeplearning.com/chap2.html.

[15] Communications, O. – Neurological Disorders, P. L. N. I. – Health Bethesda, M. . *Sleep stage* [online]. NIH National Institutes of Health - A Federal agency whose mission is to improve the health of the people of the United States. NIH is a part of the Public Health Service, which is part of the U.S. Department of Health and Human Services. Publication No. 17-3440c, 2019. [cit. 2021/12/30]. Available at: https://www.ninds.nih.gov/Disorders/Patient-Caregiver-Education/Understanding-Sleep.

[16] Ognjanovski, G. *Everything you need to know about Neural Networks and Backpropagation — Machine Learning Easy and Fun* [online]. Gavril Ognjanovski, 2019. [cit. 2022/01/04]. Available at: https://towardsdatascience.com/everything-you-need-to-know-about-neural-networks-and-backpropagation-machine-lea

[17] O'Grady, S. *The RedMonk Programming Language Rankings: June 2021* [online]. sogrady, 2021. [cit. 2022/05/03]. Available at: https://redmonk.com/sogrady/2021/08/05/language-rankings-6-21/.

[18] O'Reilly, C. et al. Montreal Archive of Sleep Studies: an open-access resource for instrument benchmarking and exploratory research. *Montreal Archive of Sleep Studies: an open-access resource for instrument benchmarking and exploratory research.* 06 2014, 7. Available at: https://doi.org/10.1111/jsr.12169.

[19] Paquette, T. *Montreal Archive of Sleep Studies (MASS)* [online]. Tyna Paquette, 2015. [cit. 2022/01/21]. Available at: http://ceams-carsm.ca/mass/.

[20] Rowe, S. *Introduction to Neurons in Neural Networks* [online]. Samuel Rowe, 2019. [cit. 2022/01/04]. Available at: https://medium.com/artificial-neural-networks/introduction-to-neurons-in-neural-networks-71828d040a65.

[21] Rychlik, J. *Author's git* [online]. Jan Rychlik, 2022. [cit. 2022/01/06]. Git Hub. Available at: https://github.com/RychlikJan/SleeepSpindlesAnnotation.

[22] Snives. *Simple Neural Network* [online]. Snives, 2018. [cit. 2022/01/04]. Available at: https://github.com/snives/SimpleNeuralNetwork.

[23] Varikuti, M. *LSTM Networks* [online]. Mlearning.ai, 2021. [cit. 2022/05/03]. Available at: https://medium.com/mlearning-ai/lstm-networks-75d44ac8280f.

[24] VERMA, Y. *A Complete Understanding of Dense Layers in Neural Networks* [online]. YUGESH VERMA, 2021. [cit. 2022/05/05]. Available at: https://analyticsindiamag.com/a-complete-understanding-of-dense-layers-in-neural-networks.

[25] Zhao, D. et al. A deep learning algorithm based on 1D CNN-LSTM for automatic sleep staging. *A deep learning algorithm based on 1D CNN-LSTM for automatic sleep staging.* 07 2021, 8. Available at: https://pubmed.ncbi.nlm.nih.gov/34180436/.

A List of abbreviations

- **XML** - eXtensible Markup Language
- **MASS** - Montreal Archive of Sleep Studies
- **REM** - Rapid-Eye-Movement
- **EEG** - Electroencephalography
- **EOG** - Electrooculography
- **EMG** - Electromyography
- **NREMS** - Non-Rapid-Eye-Movements
- **MODA** - Massive Online Data Annotation
- **PSG** - polysomnography
- **GS** - Gold Standard
- **LSTM** - Long Short-Term Memory neural network
- **CNN** - Convlutional neural network
- **EDF** - European Data Format
- **FC** - Fully Connected layer
- **NN** - Neural Network
- **SVM** - Support Vector Machine
- **GTE** - General Teting Error
- **EELR** - Euler Elastic Logistic Regression
- **RNN** - Recurent Neural Network
- **RGB** - Red Green Blue
- **Ci** - Cohort index
- **SSi** - Stage index
- **GB** - GigaByte

- **CPU** - Central Processing Unit
- **GPU** - Graphics Processing Unit
- **B-C** - Binary cross entropy

B User Document

B.1 Requirements

In the item list are named all requirements to retry the research.

- Data
  - Research description
  - Research affiliation
  - Request for the approval ethics commission
  - Approval ethics commission
  - Code for work with the data
  - Mass license
  - Data downloaded from the MASS portal
- Software
  - Windows 10+/Linux operating system
  - Git - GitHub account
  - Python 3.7
  - Conda
- Python Packages
  - MNE
  - PyEDFlib
  - Numpy
  - Pandas
  - Matplotlib
  - Torch
  - Tensorflow
  - Keras

B.2 Download and run

For a successful run the application follows these steps.

- Clone or download the project from GitHub
- Run the PyData.py with an argument *python PyData.py "Path/to/the/-folder/with/downloaded/data/C1/SS2*
- Choose and open the script for machine/deep learning and run it in cmd with the file(filepath) with data *python script.py "source64nonSpinRedForTrain.csv", "output64nonSpinRedForTrain.csv", "source64nonSpinRedToTest.csv", "output64nonSpinRedToTest.csv"*
- Wait for results

C CD Contents

The thesis is accompanied by a CD with files, closely related with the Master's thesis.

- Master's thesis
- All implemented python scripts
  - PyData.py
  - PyConvolution1DTorch.py
  - PyConvolutional1D.py
  - PyDense.py
  - PyLSTM.py
  - PyBruteForce4Channels.py
  - PyBruteForce.py
- Documents related with data request
  - Research description
  - Research affiliation
  - Request for the approval of ethics committee
  - Approval of ethics committee
  - Code for work with the data
  - Mass license
- DP-Poster-Jan-Rychlik

D Zip-file content

Listing D.1: The Readme.txt file describing the structure the resulting file.

```
[- Text_thesis
        |- MasterThesis.pdf # Text of Master's thesis
        |- pictures       # Folder with pictures
        \- source_codes # LaTex source codes

- Poster # PDF & PUB poster file

- Aplication_and_libraries       # All scripts
                                 implemented during
                                 master's thesis

- Input_data
        \-documents # All documents
                    related to the
                    data request

- Results
        \-results.xlsx  # Excel file with results
- Readme.txt
```