Utrecht University

Bachelor Thesis

Developing A Nocturnal Seizure Detector Using Support Vector Machines

Bachelor Thesis Artificial Intelligence (7.5 ECTs)
03-08-18

Author:
Alec Glasius (5718910)

Supervisor and first examiner:
M.J.C.Eijkemans

Second examiner:
M.M. Dastani

Contents

1 Problem Statement

Roughly 6000 people are diagnosed with epilepsy every year in the Netherlands alone. Approximately 70% becomes seizure free with treatment, but this still leaves a total of 120,000 people with active epilepsy [1]. One of the problems these people face is nocturnal seizures, which are seizures that occur during sleep. These nocturnal seizures can cause injury and even death when no proper care is given. This requires a caregiver to stay awake and monitor the patient the entire time he or she is asleep, which can be bothersome for both caregiver and patient. A system that detects nocturnal seizures and sounds an alarm to wake a caregiver can thus be used to reduce or prevent injury and improve quality of life for both patient and caregiver. For this reason, the Nightwatch has been made, a strap-on sensor that monitors the patient during their sleep. Using data collected from The Nightwatch, a support vector machine can be trained to possibly detect nocturnal seizures. Thus, the problem statement that guides the research of this thesis is as follows:

Is a reliable nocturnal seizure detector by training a support vector machine for the Nightwatch feasible?

Here, 'reliable' is defined as being able to predict at least 90% of the seizures, with at least 50% of the predicted seizures being actual seizures. To provide a specific answer, the problem statement breaks down into the following research questions:

1. What features do we extract from the Nightwatch's output signals?
   The Nightwatch uses four sensors to detect the heart rate, reliability of the measured heart rate, movement and angle every second. These values can be used as features directly, but other useful meta-features can be extracted as well. For example, we can add the slope of a signal to determine how much the signal increases or decreases over time.

2. How do we define the best outcome?
   Ideally, we get a 100% correct classification, but most likely this will not be the case. Thus, whenever we apply a change to the model we need to check whether it is better or worse than before. There are several ways to measure what model is best. For example, the simplest measure is the accuracy, which is the percentage of seizures correctly predicted.

3. How do we handle class imbalance?
   Our dataset suffers from a large class imbalance in seizure and nonseizure data. Support vector machines are known to perform relatively good on imbalanced data, compared to other classification models, but the imbalance might cause the model to underperform. There is no universal method to approach class imbalance, but there are several methods to ameliorate it. We should test if the support vector machine will perform better using one of these methods.

4. Should we use the kernel trick?
   Assuming the data is not linearly separable, we can still make a linear separation using a soft margin support vector machine. In order to make a more complex linear separation it is also possible to use the kernel trick. The kernel trick uses a kernel to transform the data to a higher dimensional plane, making it possible to make a (previously impossible) separation. This does, however, increase the probability of overfitting.

To provide answers to these questions, we will extract meta-features that help improve classification. Using these features, we test which measure scores models on a sample set the best and choose that one. By using the chosen measure, we can test different methods that handle class imbalance and again choose what works best. We can then test whether a kernel (and further options) will make for more improvement. Finally, we use the tested as best model to train on the complete dataset.

2 Data Collection

Prior to this project, data was collected from patients with a history of nocturnal seizures by Stichting Epilepsie Instelling Nederland (SEIN). Before a patient went to bed, a Nightwatch sensor was attached to the upper arm of the patient. Using a heart rate and three axis motion sensors, the Nightwatch recorded four base features (table 1) roughly every second during the night. Seizures were then identified and marked by hand on the correct timestamps using video footage. This resulted in a total of 3510 hours of footage covering 1907 nights of 44 patients overall (of whom 35 had seizures) and containing 1682 seizures, of which 1112 clinically urgent.

Table 1 Example of base features

HR	Conf	Motion	Angle
88	70	5	-3

HR heart rate in beats per minute (BPM); Conf confidence in measured heart rate in range (0 – 100);
Motion back and forth movement in Hertz (Hz); Angle roughly in range (-16 – 16)

2.1 Preprocessing

To make the data ready for training, we aggregate it from seconds to minutes in order to downsize the dataset and reduce the computational load. The data reduced from 1.17e+8 to 1.98e+6 data points after aggregation.

Second, the data seizures are marked as single points, but we count a seizure as successfully predicted if at least one point within 5 minutes before or after is predicted as seizure. Therefore, we label the complete 10 minute time frame as seizure point in our dataset. This also greatly reduces the class imbalance; from 1:1775 to 1:185, leaving us with roughly 1.06e+4 seizure points.

Finally, we split the data in a training and test set. The training and test set are made by randomly assigning all data from a patient (stratified by the number of patients with and without seizures) to either training or test set. This results in the training set containing 9.99e+5 data points from 23 patients with a total of 5.37e+2 seizures and 5.17e+3 seizure points. The test set contains 9.76e+5 data points from 21 patients with a total of 5.75e+2 seizures and 5.47e+3 seizure points. From these two sets, we extract a train and a test sample, each containing all data from nights with at least 1 seizure, summing to a total 50 seizures. We perform all testing on these samples in order to reduce computational load. All aforementioned and following actions throughout this thesis are computed in R.

3 Feature Extraction

Given the base feature signals, we can extract more information from the changes in each signal. We extract new meta-features based on the physiology of seizures and common feature extraction methods from signal processing (table 2). The majority of seizures come with a rise in heart rate and either a rise in random movement or in a specific frequency (2 – 6 Hz). We use this information to create new features that help separate seizures from normal sleep.

3.1 Slope

As mentioned, seizures cause a rise in heart rate and movement. To identify this process, we calculate the slope of both the heart rate and movement signals. We expect that the slope of both signals is high when seizures start, low when it ends and near 0 otherwise. We calculate the slope using the following linear regressing formula:

$$b = \frac{MEAN(x) * MEAN(y) - MEAN(x * y)}{MEAN(x)^2 - \ MEAN(x^2)}$$

Here, y is the signal's value on each point in the last 10 minutes and x the corresponding indexes (index being the nth measured value in a night). The slope was only calculated for the heart rate and motion signals since a slope for the angle would mainly result in an indication of movement, which is already present in the motion feature [2].

3.2 Cumulative Sum

We use a second measure to identify the rise in heart rate and movement, the cumulative sum. We not only expect seizures to cause a rise in both the signals, but also to rise above any normal value that can be expected during sleep. We use the cumulative sum as it takes the average signal value into account. It is also better at identifying signals that rise slower over a longer period of time than the slope is. The cumulative sum of a signal is calculated by taking the previous cumulative sum (as long as it is higher than 0) plus the current signal, minus the average of the signal so far [2].

$$CUMSUM_n = MAX(CUMSUM_{n-1}, 0) + (x_n - MEAN_{1:n}(x))$$

Here, n is the signal index. This means the cumulative sum can only exceed zero when the signal exceeds the average and continues to rise as long as the signal is above average, thereby taking the duration of the signal change into account. We expect the cumulative sum to increase when seizures cause the heart rate and movement signal to rise above normal values. Otherwise we expect it to stay near 0 during sleep, when the signals are more stable. In contrast to most other features, the cumulative sum is calculated with all previous data from the current night, instead of with the previous 10 minutes. It is again not calculated for the angle signal as this would also mainly result in an indication of movement, which is already present in the motion feature.

3.3 Variance

During seizures, the heart rate, motion and angle signal will be less stable than during sleep. To keep track of how stable a signal is, we use the variance. The variance is a measure of dispersion of data and is defined and calculated as the average squared deviation from the mean [9].

$$\sigma = \frac{\sum(X - \ \mu)^2}{N - 1}$$

We calculated the sample variance from the last 10 minutes of the heart rate, motion and angle signals. Because seizures cause all three signals to become less stable, we expect a rise in variance during a seizure. The motion variance might, however, stay low if the patient moves rhythmically in a specific frequency for a long time.

3.4 Autocorrelation

During sleep, we do not expect any sudden changes in signal values. This makes for stable predictable signals, which is not the case during seizures as these cause more random signals. To represent this, we added the autocorrelation. The autocorrelation is defined as the covariance with a delayed copy of itself [8].

$$R(\tau) = cov(X_t, X_{t+\tau})$$

Here, τ is a time lag. We used a time lag of 5 minutes to calculate the autocorrelation of the heart rate, motion and angle signal. For each signal, we compared the previous 5 minutes to the 5 minutes preceding those. If both the signal and the delayed signal show similar behavior (that is increase or decrease accordingly) the autocorrelation will be positive. If both signals show opposite behavior (when one tends to be high when the other is low) the autocorrelation will be negative. One disadvantage is that the autocorrelation might increase if movements in a specific frequency persist continuously as this also makes for a predictable signal.

3.5 Shannon Entropy

As mentioned the signals are more likely to be stable during sleep and more random during seizures. To better represent the randomness of a signal that we expect during seizures we use the Shannon entropy. Shannon entropy provides a way to estimate the average minimum number of bits needed to encode a signal, based on the frequency of the signals values [7]. It is given by the formula:

$$H(X) = -\sum_i p_i log_2(p_i)$$

Where p_i is the probability of a value at signal index i. In information theory, entropy is a measure of unpredictability of a state.

3.6 Daily risk

Besides the immediate physiological changes during a seizure, there are several triggers that can increase the risk of a seizure. Some of these triggers are sleep deprivation [3], increased stress [4] and hormonal changes during the menstrual cycle [5]. To represent these periods of increased risk, we added two features that are calculated once a day instead of every minute. To represent sleep deprivation, we added the average time slept during the previous week (in seconds) as feature. From our data, it is impossible to derive the other named risk factors, but since all these factors lead to more seizures, we can use the amount of previously measured seizures as an indication of high risk periods. Thus, we added the average amount of seizures during the previous week as features to represent the other risk factors that cannot be derived directly from our data.

Table 2 Example of meta-features

HR Slope	HR Cumsum	HR Variance	HR Autocorrelation	HR Entropy
-0.88	-0.06	8.51	-0.40	9.22

Motion Slope	Motion Cumsum	Motion Variance	Motion Autocorrelation	Motion Entropy
-0.96	-0.38	0.33	-0.05	5.34

Angle Variance	Angle Autocorrelation	Angle Entropy
58.02	-9.91	8.92

Avg Sleep	Avg Seizures
37660	0.20

HR hearth rate; Cumsum cumulative sum; Avg average

3.7 History Components

Most of our new features try to separate seizures from sleep by focusing on a specific pattern in either one (and the lack of that pattern in the other). Calculating a specific measure shows this pattern as a number. In order to let the model figure out more patterns by itself, we added the four base features as history components. The history component consists of five extra features per base feature [2]. Each new feature is the value of corresponding base feature at 2, 4, 6, 8 and 10 minutes earlier.

Table 3 Example of hearth rate history components

HR 2 minutes earlier	HR 4 minutes earlier	HR 6 minutes earlier	HR 8 minutes earlier	HR 10 minutes earlier
84	80	81	78	80

HR hearth rate

3.8 Results

As most meta-features require at least 10 minutes of data, all nights lasting less are removed before calculating meta-features. After meta-feature calculation, the first 10 minutes of data of each night are removed as these do not have values for all meta-features. This reduced the dataset to roughly 1.89e+6 points containing 1.03e+4 seizure points, with every seizure point having 39 features. All features are then scaled using standardization in such a way that features from the training set have a mean of 0 and a standard deviation of 1.

4 Scoring

Before we can start testing different methods and settings, we need to decide how we score a support vector machine model. To decide on a good scoring measure, we train three models on the training sample using 10-fold cross validation to find the optimal parameter values (see 6.2) where we keep all folds identical for training the different models. Each model uses one of the below named measures as scoring function. We then use each model's predictions during cross validation to calculate the ROC curve and the area underneath the curve (AUC). The AUC is a way to evaluate a binary classification model. We choose the measure that makes for the highest AUC as the optimal measure (table 4).

4.1 Accuracy

The simplest measure is the accuracy, the percentage correctly predicted points. To calculate the accuracy on predicted data, we first follow our criteria for when a seizure is successfully predicted. We count all true negatives (TN) and false positives (FP) from the nonseizure data. For seizure data, we count one true positive (TP) if the 10 minute time frame contains at least one point predicted as seizure and counts one false negative (FN) otherwise. All other points in the seizure are counted as true negatives. The accuracy is then calculated with:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

The accuracy is very sensitive to class imbalance, because predicting every point as the majority class is often the easiest way to maximize the accuracy, resulting in a high accuracy without correctly classifying a single point from minority class.

4.2 F-Measure

Another measure that is less sensitive to class imbalance is the F-Measure. To calculate the F-Measure, we again follow our criteria for when a seizure is successfully predicted, done the same way as with accuracy. Now we look at the percentage of seizures that are successfully predicted as seizures (recall) and the percentage of predicted seizures that are real seizures (precision). We then combine these two measures as the recall.

$$precision = \frac{TP}{TP + FP} \qquad recall = \frac{TP}{TP + FN} \qquad F\ Measure = \frac{2 * precision * recall}{precision + recall}$$

Not only is the large amount of TNs that we have because of the class imbalance now disregarded, precision and recall also correspond exactly to our 'reliable' detector criteria. This means we need a recall of at least 0.9 and a precision of at least 0.5 to have a reliable detector.

4.3 Log Likelihood

The log likelihood is a measure that is affected even less by class imbalance. That is because it uses class probabilities instead of hard classifications, contrary to the accuracy and F-Measure. To calculate the log likelihood, we sum up the logs of the class probability to see if it is the correct class .

$$Log\ likelihood = \sum_{i=1}^{n} y_i \log\big(p(x_i)\big) + (1 - y_i)\log(1 - p(x_i))$$

Here $p(x_i)$ is the probability that the ith point is a seizure and y_i is its class (0 or 1). Because the log likelihood uses class probabilities instead of hard classifications we cannot use our criteria for a successfully predicted seizure.

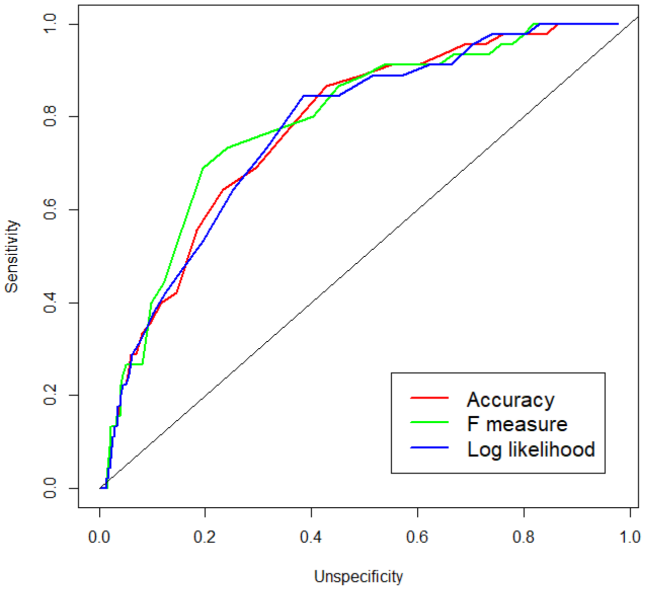
4.4 Results

To calculate the ROC curve of each model (figure 1), we use the class probabilities predicted during cross validation. Using these class probabilities, we calculate the sensitivity and unspecificity on all thresholds between 0 and 1 in steps of .001.

$$Sensitivity = \frac{TP}{TP + FN} \qquad Specificity = \frac{TN}{TN + FP} \qquad Unspecificity = 1 - Specificity$$

The closer the line is to the top left corner the higher the sensitivity and specificity are, and thus the better the model is, the closer it is to the diagonal line, the worse the model is.

Figure 1 ROC curves of models trained with different measures



ROC Receiver operating characteristic

We can then calculate the exact area under each curve, in order to know exactly which model is the best. As we can see in table 4, the F-measure scores the best with an AUC of .7657. From here on we will use this measure. In table 4 we can also see that accuracy, surprisingly, performs better than the log likelihood. This might be because the support vector machine is by nature a hard classification model.

Table 4 AUC score for different measures

Measure	AUC
Accuracy	.7517
F measure	.7657
Log likelihood	.7487

AUC Area under the curve

5 Class Imbalance

Our dataset has a very heavy imbalance in seizure and nonseizure data. This could cause the model to classify too many points as the majority class (nonseizure). There is no universal method to approach class imbalance. Many methods work on some datasets, but not on all. We test several of these methods to see if they have a positive effect on our dataset and choose the best to use in further training. The best is here defined as the one with the highest F-measure, as concluded in 4.4. To do this, we first extract complete nights with a total of five seizures from the training sample and set these aside for calibration. We tune each model using 10-fold cross validation to find optimal parameter values (see 6.2) for the (combinations) of the methods named below and test these on the test sample (table 5). We test the methods separately for models without kernel and for models with a Radial Basis Function kernel (RBF kernel, see 5.2), as these methods might perform different on each.

5.1 Class Weighting

A simple method to deal with class imbalance is to introduce an unequal cost ratio on the confusion matrix during model training. This is done by applying a higher weight on false negatives (equal to the imbalance ratio) than on false positive. This way, the model will try to misclassify true positives less often, at the expense of misclassifying true negatives more often.

5.2 Under/Over Sampling

Another option is to balance the training data by either undersampling the majority class or oversampling the minority class. This is done, respectively, by randomly removing nonseizure data (figure 2) or duplicating seizure data (figure 3) until both classes are balanced. While both make the data balanced, they each have their disadvantages . By undersampling, we discard a large portion of useful information, which is bad in itself and also causes more variability in results. By oversampling, we duplicate the seizure data, which also results in duplicating any false negative errors. This means that when the model makes one false negative error, it counts multiple errors, making it more likely for the model to overfit in order to prevent as much false negatives as possible. It also greatly increases the computational load as there simply are a lot more data points.

Figure 2 Under sampling majority class

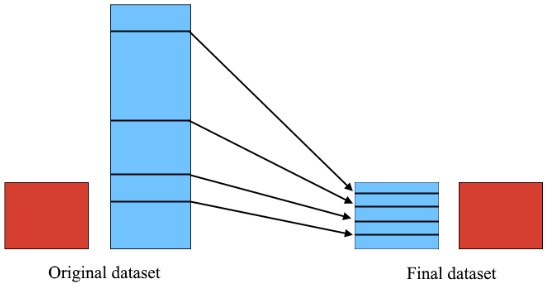
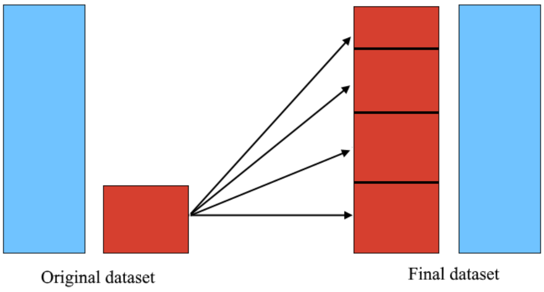


Figure 3 Over sampling minority class



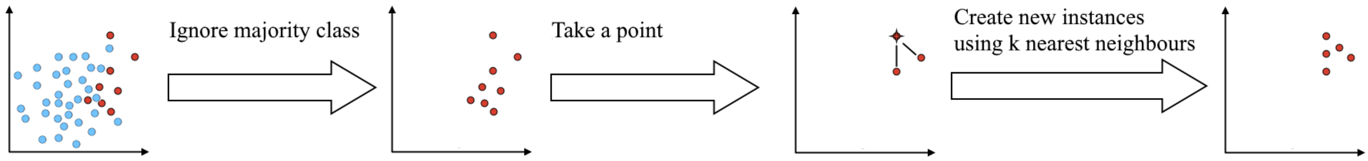
5.3 Under Sampling + Bagging

The weakness of undersampling is the discarding of data and having fluctuating results when training on the same data set. To try and make up for this we can choose to apply bagging [6]. That is training several models using undersampling and using a majority of vote to classify a point. Doing so not only causes us to discard less data, it also lowers the variance, leading to steadier results. It does, however, cause an increase in classification time as several models need to make a prediction. For our dataset, we tested a combination of five models, as this number of combinations still has a reasonable prediction time.

5.4 SMOTE

SMOTE is a technique similar to oversampling, but instead of duplicating the minority class, it creates new points. It makes these points by ignoring the majority class and taking a minority class point and its k nearest neighbours. Then, it creates new instances of the minority class somewhere on the line between the point and each of the k neighbours (figure 4). Doing so, results in a balanced dataset that is less likely to overfit than plain oversampling.

Figure 4 Flow chart of SMOTE on imbalanced data



SMOTE Synthetic Minority Oversampling Technique

5.5 Optimal Class Probability Threshold

The support vector machine will separate the data, using a hyperplane, and classify all points on one side as seizures and on the other as nonseizures. The closer a point is to this hyperplane, the less confident the model is in its prediction. This confidence is the class probability. We can find a threshold which instructs the model that it needs a certain level of confidence that a point is a seizure, before classifying it as such. The threshold that gives the best F-measure for this is the optimal class probability threshold.

To find this optimal threshold, we can make the model predict in probabilities instead of hard classifications. However, these probabilities are not real probabilities as they are uncalibrated, meaning that they might not use the full 0-1 range and that the difference between 0.1 and 0.2 might not be the same as the difference between 0.2 and 0.3. To calibrate the probabilities, we use Platt scaling. This means we first let the tuned support vector machine predict the class probabilities on the calibration data we set aside. Given the probability that points are seizures, we train a logistic regression model to predict the class. We can then use this calibration model to predict the probability a point is a seizure given the support vector machines probability it is a seizure. By doing so, we refine the class probabilities and calibrate them.

Now that we have the calibrated probabilities, we can test different thresholds stepwise from 0 until 1 in steps of .001 and choose the one that results in the best F-measure. In order to test our model on test data, we must first calibrate it again using the calibration model we found and then apply the found threshold to classify points.

5.6 Results

We use the F-measure to test class weighting, undersampling, oversampling, undersampling + bagging and SMOTE, both with and without optimal threshold and separately for models with and without RBF kernel (table 5).

For both the models with RBF kernel and without, not a single method increases the F-measure compared to using no method at all. This could be because all methods simulate a balanced training set in some way, causing the model to expect more seizures during testing than are the case. Using the optimal threshold on calibrated class probabilities could lower this expectation and cause for a better model. Doing so causes an increase in F-measure in half of the models, but none are better than using no method.

As can be seen in table 5, using no method gives the best F-measure resulting in an F-measure of .4810 for models with RBF kernel and .6154 for models without kernel. Thus, in the next step of training, the support vector machine model with no extra methods to handle class imbalance will be used.

Table 5 Results of different methods for class imbalance

Method	Threshold used	F-measure	F-measure (RBF)
None	0	.6154	.4810
	1	.2989	.4643
Class weighting	0	.1331	.2640
	1	.1914	.1654
Undersampling	0	.5435	.0390
	1	0	.2201
Oversampling	0	.1162	.0218
	1	.1300	.0127
Undersampling + Bagging	0	.5581	.0413
	1	.1650	.2036
SMOTE	0	.1905	.0424
	1	.2533	.1523

RBF Radial Basis Function

6 Training

Before training the support vector machine we want to find the best way to train it. To do this we again use the sample data to train and score several options. We do this by testing all combinations of the below named options on the sample training set, using 10-fold cross validation where all folds are kept identical for every option (table 6). We do not use any extra methods to handle with the class imbalance, as concluded in 5.6. The best combination of options will then be used to train and test on the complete dataset.

6.1 Further Options

Further training options are available. First, there is the option of using a kernel. We choose to use the RBF kernel as it is a widely used, relatively easy to tune, effective on imbalanced data and proven useful in very similar research [2]. Then there is the option to not train on seizures as a 10 minute timeframe and keep them as a single point. This will result in an even more unbalanced dataset, but only data from the seizures peak will be used, possibly making for a better separation with nonseizure data. Lastly there is the option of aggregating after meta-features are added instead of before. Less patterns might be lost by doing so resulting in better predictions, but this is also more computation heavy.

6.2 Parameters

The C parameter is the cost of misclassification. A high C value will result in a smaller margin hyperplane (if that smaller hyperplane will make for more correctly classified observations), which can result in an overly complex and over fitted model. Contrary to this, a low C value will make for larger margin hyperplanes, even if this will result in more misclassified observations. Tested C values are: $10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}$ and 10^{1}.

The γ parameter is only used by the RBF kernel and indicates how complex the kernel's transformation is. A large γ will again increase the risk of overfitting while a low γ may fail to find any existing patterns. Tested γ values are: $10^{-3}, 10^{-2}, 10^{-1}, 10^{0}$ and 10^{1}.

6.3 Results

As can be seen in table 6, the best result, with an F-measure of .5283, is obtained using the RBF kernel, training on seizures as time frame, aggregating after adding meta-features, using a C value of 10 and γ value of .01. Furthermore, we can see using the RBF kernel improves every model. Using seizures as single points greatly worsens every model and aggregating after meta-features worsens the model only in one case.

Table 6 Cross validation results of different training options

Without/ With RBF kernel	Seizures as time frame/ point	Aggregate before/ after meta-features	C	γ	F measure
without	time frame	before	1	-	.2588
without	time frame	after	.1	-	.2787
without	point	before	10	-	.0855
without	point	after	.1	-	.0143
with	time frame	before	10	.01	.4781
with	time frame	after	10	.01	.5283
with	point	before	1	.1	.0969
with	point	after	.1	.1	.1308

Using the combination of options from the just named best model we train the model on the entire training set and test it on the test set (table 7). With this we unfortunately did not reach the goal of a 0.9 recall and 0.5 precision.

Table 7 Best model trained and tested using all data

Recall	Precision	F measure
.6743	.3159	.4303

7 Conclusion

To answer the problem statement, “Is a reliable nocturnal seizure detector by training a support vector machine for the Nightwatch feasible?”, that guides this thesis we first extracted meta-features based on the physiology of seizures and common feature extraction methods from signal processing that help separate the seizure data from the non-seizure data. We tested several scoring measures on a train sample and concluded the F measure as the best, which is also in line with our definition of reliable. Using the F measure we tested multiple methods on a train and test sample that can improve models trained on imbalanced data but used none as all methods resulted in a worse performance. Again using the F measure and a train sample, we tested multiple training options to find the best model which resulted in aggregating the data after adding meta-features and the use of the RBF kernel. Finally, we used these training options to train a model on the complete training data and tested it on the test data which resulted in a recall of 0.6743 and a precision of 0.3159. This did not reach our goal of at least a recall of 0.9 and precision of 0.5, meaning it was unfortunately not feasible to make a reliable nocturnal seizure detector by training a support vector machine for the Nightwatch.

For further research other methods that handle class imbalance can be tested that possibly increase performance. Besides using a support vector machine models, other classification models can also be tested for better performance on this problem. A combination of several classification models, using an ensemble method, can then also be used to further improve performance.

8 References

1. Epilpesiefonds. Retrieved from: https://www.epilepsie.nl
2. Andel van. J. Towards Multimodal System for Nocturnal Seizure Detection.
3. Beth A. Malow M.D. Sleep Deprivation and Epilepsy. doi:10.1111/j.1535-7597.2004.04509.x
4. Nancy R. Temkin, Gay R. Davis. Stress as a Risk Factor for Seizures Among Adults with Epilepsy. doi: 10.1111/j.1528-1157.1984.tb03442.x
5. Steven C. Schachter. Hormonal Considerations in Women With Seizures. doi:10.1001/archneur.1988.00520350105025
6. Wallace, Small, Brodley and Trikalinos. Class Imbalance, Redux. doi:10.1109/ICDM.2011.33
7. Kumar Y., M.L Dewal. Complexity Measures for Normal and Epileptic EEG Signals using ApEn, SampEn and SEN. International Journal of Computer & Communication Technology, 2(7), 6-12
8. Chan PH, Wong CK, Poh YC, et al. Diagnostic performance of a smartphone-based photoplethysmographic application for atrial fibrillation screening in a primary care setting. https://doi.org/10.1161/JAHA.116.003428
9. Sassi R. et al. Advances in heart rate variability signal analysis: joint position statement by the e-Cardiology ESC Working Group and the European Heart Rhythm Association co-endorsed by the Asia Pacific Heart Rhythm Society. doi:10.1093/europace/euv015