COMENIUS UNIVERSITY IN BRATISLAVA
FACULTY OF MATHEMATICS, PHYSICS AND INFORMATICS



CLASSIFICATION OF RAT STRAINS USING MACHINE LEARNING

MASTER'S THESIS

2021 Bc. Eduard BUČKO

COMENIUS UNIVERSITY IN BRATISLAVA
FACULTY OF MATHEMATICS, PHYSICS ANDINFORMATICS

CLASSIFICATION OF RAT STRAINS USING MACHINE LEARNING

MASTER'S THESIS

Study programme:	Mathematics of Economy, Finance and Modeling
Branch of study:	1113 Mathematics
Department:	Department of Applied Mathematics and Statistics
Supervisor:	Mgr. Katarína Boďová PhD.

Bratislava 2021 **Bc. Eduard BUČKO**

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

KLASIFIKÁCIA KMEŇOV POTKANOV POMOCOU STROJOVÉHO UČENIA

DIPLOMOVÁ PRÁCA

Študijný program:	Ekonomicko-finančná matematika a modelovanie
Študijný odbor:	1113 Matematika
Školiace pracovisko:	Katedra aplikovanej matematiky a štatistiky
Vedúci práce:	Mgr. Katarína Boďová PhD.

Bratislava 2021 **Bc. Eduard BUČKO**





Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

THESIS ASSIGNMENT

Name and Surname:	Bc. Eduard Bučko
Study programme:	Mathematical Economics, Finance and Modelling (Single degree study, master II. deg., full time form)
Field of Study:	Mathematics
Type of Thesis:	Diploma Thesis
Language of Thesis:	English
Secondary language:	Slovak

Title: Classification of rat strains using machine learning

Annotation: Rats are commonly used model animals, used for instance to study circadian rhythms and the impact of light regime on them. To record behaviour of the animals, telemetry recording methods are often used. The outcome is in the form of the well-resolved dataset of the key biological characteristics of the animal over the extended time (days, weeks). The main goal of this work is to use the recorded data (experiments done by Ľ. Molčan, PRIF UK, Bratislava) to identify the rat strain, i.e., infer the genotype from the phenotype. Methods from the area of machine learning will be used to solve the classification problem. One of the goals will be to find out how much data (and how granular) is needed for a sufficiently accurate classification.

Supervisor:	Mgr. Katarína Boďová, PhD.
Department:	FMFI.KAMŠ - Department of Applied Mathematics and Statistics
Head of department:	prof. RNDr. Marek Fila, DrSc.
Assigned:	17.12.2019
Approved:	21.12.2019

prof. RNDr. Daniel Ševčovič, DrSc.
Guarantor of Study Programme

.....
Student

.....
Supervisor





Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Eduard Bučko
Študijný program: ekonomicko-finančná matematika a modelovanie (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: matematika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: anglický
Sekundárny jazyk: slovenský

Názov: Classification of rat strains using machine learning
Klasifikácia kmeňov potkanov pomocou strojového učenia

Anotácia: Potkany sú bežne používané modelové experimentálne zvieratá, napríklad pri štúdiu cirkadiánnych rytmov a vplyvu svetelného signálu na tieto rytmy. Správanie potkanov sa v súčasnosti zaznamenáva pomocou tzv. telemetrických metód, ktoré produkujú vysokofrekvenčný plynulý záznam dôležitých biologických charakteristík zvieraťa počas dlhšieho obdobia (dni, týždne). Hlavným cieľom práce je na základe zaznamenaných dát (dáta pochádzajú od Ľ. Molčana, PRIF UK, Bratislava) identiikovať typ potkana, teda z fenotypu potkana určiť jeho genotyp. Klasifikácia sa prevedie metódami z oblasti strojového učenia. Súčasťou práce je aj zistiť, koľko dát (a ako podrobných) je potrebných na klasifikáciu s uspokojivými výsledkami.

Vedúci: Mgr. Katarína Boďová, PhD.
Katedra: FMFI.KAMŠ - Katedra aplikovanej matematiky a štatistiky
Vedúci katedry: prof. RNDr. Marek Fila, DrSc.

Dátum zadania: 17.12.2019

Dátum schválenia: 21.12.2019 prof. RNDr. Daniel Ševčovič, DrSc.
garant študijného programu

.....
študent

.....
vedúci práce

Acknowledgements

This way, I would like to express my sincere gratitude to Mgr.Ľuboš Molčan, PhD. for the provided motivation, scientific consultation and the data used in this study.

Above all, I would like to thank Mgr. Katarína Boďová, PhD. for her excellent explanations of the studied material, prompt responses and untiring patience.

Abstrakt

BUČKO, Eduard: Klasifikácia kmeňov potkanov pomocou strojového učenia [Diplomová práca], Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky, Katedra aplikovanej matematiky a štatistiky; školiteľ: Mgr. Katarína Boďová PhD., Bratislava, 2021, 50 str.

Cieľom našej práce bolo nájsť rýchly a efektívny spôsob klasifikácie štyroch rôznych kmeňov potkanov (dva hypertenzné a dva normotenzné), s použitím strojového učenia. Na učenie klasifikačného modelu sme použili telemetrické dáta potkanov získane v experimente Molčan et al. Požili sme tri metódy na štúdiu dát v troch stupňoch granularity. Ako prvé boli porovnané dlhodobé (12 hodinové) priemery biologických ukazovateľov, t.j. krvný tlak, aktivita, srdcová frekvencia. Ako druhé sme na klasifikáciu použili momenty 5-minútových beat-to-beat dát. Tretia metóda mala za ciel zlepšiť výsledky klasifikácie v porovnaní s predošlými hrubšími metódami. Zameriavala sa na jemné vlastnosti zaznamenaných dát. Použili sme pri tom metódu rekonštrukcie atraktora založenej na trojdimenzionálnej projekcii oneskorení časového radu. Analyzovali sme pri tom, či umožňujú vlastnosti tvaru vlny krvného tlaku klasifikovať kmene potkanov v takmer reálnom čase. Metódy boli schopné rozlíšiť medzi normotenznými a hypertenznými potkanmi, a dokonca aj rozlíšiť kmene vrámci hypertenznej skupiny. Metódy nesprávne určili niekoľko normotenzných potkanov, ale celkovo boli výsledky úspešné. Vynikajúco dopadli najmä prvá a posledná z použitých metód.

Kľúčové slová: Rekonštrukcia atraktora, Support vector machine, Automatická klasifikácia

Abstract

BUČKO, Eduard: Classification of rat strains using machine learning [Diploma thesis], Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, Department of Applied Mathematics and Statistics; Supervisor: Mgr. Katarína Boďová PhD., Bratislava, 2021, 50 pg.

This study aims to find a quick and effective way of classifying four different strains of rats (two hypertensive and two normotensive strains) with machine learning. To learn classification models, we used telemetry data of rats, obtained in the experiments of Molčan et al. Three methods are employed to study the data on three different levels of granularity. Firstly, the long-term (12 hours) averages of biomarkers, i.e., blood pressure, activity, and heart rate, were compared. Secondly, the moments of five-minute beat-to-beat data were used for classification. The third method aims to improve the classification results of the coarser methods by looking at the fine properties of the recorded data. It uses an attractor method based on a 3-dimensional projection of the delayed time series. The fine-scale features of arterial blood pressure waveforms were analysed to see if the rats could be classified near real-time. The methods could differentiate between normotensive and hypertensive rats and even assign the correct of the two strains to hypertensive rats. The methods misclassified some normotensive rats, but overall the results were successful. The first and the last methods scored especially good.

Keywords: Attractor reconstruction analysis, Support vector machine, Automatic classification

Contents

1 Introduction **11**

1.1 Research Questions 11

1.2 Approach and Methodology 12

1.3 Scope and Limitation 13

1.4 Target Group 13

1.5 Outline 13

2 Theory **15**

2.1 Supervised Learning 15

2.1.1 Data Pre-Processing 15

2.1.2 Algorithm Selection 17

2.2 Support Vector Machine 18

2.2.1 Maximal Margin Classifier 19

2.2.2 Support Vector Classifier 20

2.2.3 Kernel Trick 21

2.2.4 Generalisation for Multi Class Problem 22

2.3 Overfitting 23

2.3.1 Train-Test Split 23

2.3.2 Cross-Validation 23

2.3.3 Grid-Search 24

2.4 Attractor Reconstruction Analysis 24

2.4.1 Reconstructing the Attractor 25

2.4.2 Removing Baseline Variation 26

2.4.3 Constructing a Density 26

2.4.4 Choosing the Right τ 26

3 Data **30**

3.1 Aggregate Data 30

3.2 Beat-to-Beat Data 32

3.3 Waveform Data 32

4 Applications **35**

4.1 Classification Using Dark/Light Phase Averages 35

4.2 Classification Using Beat-to-Beat Data 39

4.3 Classification Using Attractor Method 41

Conclusion **46**

Appendix **51**

1 Introduction

Using automatic measuring instruments enables gathering a large amount of data with relatively low effort. One of the methods enabling long term gathering of high-frequency data is telemetry. The analysis of a large amount of data is complex and calls for automating the analysis methods. The first condition of the analysis is grouping individuals into groups. Automatic clustering and data analysis are the future of modern experimental research. Automated systems enable not only to classify individuals but also explore new relationships between the variables.

The aim of the study is to explore the effectiveness of machine learning when classifying rat strains. Rat strains represent a group of rats with uniform genes. Different strains are created through inbreeding or other artificial processes. These specially bred strains are often used in laboratory experiments, because rats serve as a good animal model for many of research questions [1]. The task of assigning a strain to a rat is often done manually and using only rough averages. Trained models give us a tool to instantly identify the strain after we are finished collecting the required data.

We consider four strains of rats:

- Wistar Rats (*WT*)
- Spontaneously Hypertensive Rats (*SHR*)
- Hannover Sprague-Dawley Rats (*HanSD*)
- The Transgenic TGR (mREN-2)27 Rats (*TGR*)

Both *SHR* and *TGR* rats are hypertensive, meaning that they have persistently elevated blood pressure in adulthood. SHR are a result of generations of selectively breeding rats that display fenotypes of hypertension. On the other hand, TGR strain was achieved by applying transgenic techniques [2]. *WT* and *HanSD* are normotensive rat strains. [3][4]

1.1 Research Questions

Our primary goal is to find a method, which would be accurate. Classifying rat strains is one of the first tasks needed before conducting further research. There is only a tiny

margin of error allowed in research. Misclassification error could further go unnoticed and result in false laboratory experiment outcomes.

Throughout the whole process of choosing the suitable method and evaluating the results, the primary question we ask ourselves is:

- How accurately can the trained models predict strains of rats?

Parallel to the accuracy, another aspect of the method performance will be monitored. Time is a valuable resource, and as such, it is vital to minimise the time it takes for a method to classify the strain correctly.

The time it takes for a trained classifier to predict the expected genome is negligible for all the methods used in this thesis. Of primary concern is the time needed to gather the necessary data. Thus the next question asked is:

- How much time is needed for data collection when employing the method to classify new rat?

Methods using various levels of data granularity are considered in this thesis. A day, few minutes, or few seconds of rat monitoring are needed, depending on the method explored.

The current commercially available software does not deal with the problems presented in this work. Creating a good classifier would fill this hole in the market. A possible implementation of the methods into the already existing telemetry software would undoubtedly be valuable and most probably sought after.

1.2 Approach and Methodology

Machine learning is chosen in this work as a process to construct a classification model. With the help of modern machine learning algorithms, more nuanced differences can be uncovered. Machine learning also helps with defining statistically optimal boundaries between the rat strains in the space of biosignals.

First, in the research process, the available data were examined. The datasets were explored to find candidate information that will further serve to form a classification model. Next, the validation process and evaluation process were defined. Finally, suitable

methods were selected. That includes defining the data pre-processing techniques, data structures and machine learning algorithms. This part of the process was done iteratively, i.e., selecting one method, evaluating it and adjusting accordingly.

After deciding on the method used, the necessary software was selected. For data handling and to produce initial statistics, SQL was used. The rest of the process was implemented in Python. Additional Python libraries were chosen to reduce the time and effort needed to complete this work.

The results were analysed and compared. In addition to standard *Support Vector Machine* method, we also used *Attractor Reconstruction* method. Their usefulness was considered, and future possibilities were drafted.

1.3 Scope and Limitation

To accommodate deadlines, only a few machine learning methods were used. Furthermore, the two more granular datasets were limited to a reduced sample of rats. The research was constrained with storage capacity and computational power. Those will be necessary to handle larger datasets and further validate the models.

1.4 Target Group

Researchers can implement some of the methods to streamline the process of rat classification for laboratory experiments. In future works, classification could be used to analyse individual rats that were given treatment and study the effects of the treatment.

This work could inspire further research. To improve the methods used on these or different rat strains; To experiment with machine learning methods on different animal species.

1.5 Outline

First, we will present the essential theory needed. A brief outline of the machine learning pipeline will be presented before mentioning the specific methods. We will succinctly describe the Support Vector Machine algorithm used for classification and the attractor reconstruction method that we will later use to construct new features for the machine

learning model training. In the section after that, we will present the available data. In the last section of the thesis, the theory is applied to the datasets. The methodology will be described, and the final results are presented and discussed.

2 Theory

In this chapter, the essential theoretical background is described. We start with defining the techniques used in the machine learning process. Later, the support vector machine algorithm is described. The chapter ends with the theory of the attractor reconstruction method.

2.1 Supervised Learning

Supervised learning is a process in which we give computer a matrix X of input variables called features and a vector y of output variables called labels. We also choose an algorithm that will be used so that the computer learns a function $\hat{f}$ which approximates a function $f(X) = y$. This function maps the features onto labels. Our goal is to find and validate a method, which inputs new unlabelled features into the newly created function $\hat{f}$ and gives as an output a good prediction of what the label should be for those features.

The process of applying supervised ML is described in [5]. The diagram showing the process is in Figure 1.

The general pipeline of the considered methods starts with defining a problem. One starts with an interdisciplinary problem, for example, identification of objects (e.g. apples) in a picture, identification of distinct types in a population, etc. The desired type of outcome (positions of objects, particular clustering of individuals, etc.) has to be specified before applying the methods.

Then the dataset needs to be collected. It should be a sample of the population that is to be labelled. Ideally, the dataset contains descriptive attributes of the sample points. For supervised learning, sample points must be labelled in the starting dataset.

2.1.1 Data Pre-Processing

Data pre-processing involves multiple methods to deal with complex data and prepare the dataset for a machine learning algorithm. Methods included in [6] are:

- *Instance selection.* When the dataset is too large to employ machine learning algorithms effectively, a subset of the original dataset can be chosen. *Random sampling* is one of the methods that can be used. An alternative is *stratified sampling*, which

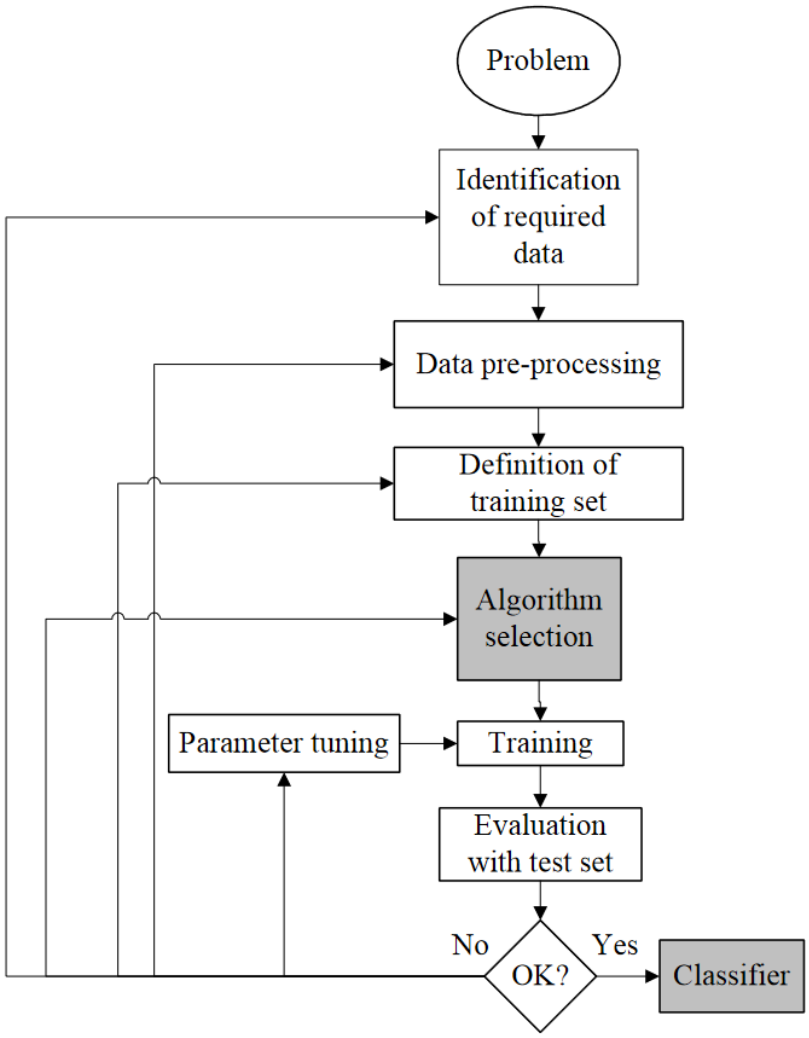


Figure 1: Diagram showing the steps of the supervised learning process. Source: [5].

is preferred if the dataset contains minority classes. In that case, random sampling could by chance completely omit one of the classes from the subset. In this work Random sampling was used to select training sets and test sets.

- *Outliers detection.* Due to machine or human error, or because of extreme conditions, some sample points can contain erroneous data. These sample points are called outliers and are generally unwanted in the machine learning process. Some algorithms are sensitive to outliers, and including them in the analysis would yield spurious results. Statistical methods of various complexity are used to find outliers in large datasets [7]. A data point is considered an outlier if it is deemed too extreme compared to the rest of the data points. Outlier values should be removed,

and missing values should be dealt with as described in the next point.

- *Handling missing feature values.* In case of a feature value missing for one of the instances, few methods are proposed. The first option is to ignore the sample point completely. If the dataset is large enough and the analysis allows it, this is the most easiest solution. The other option is to replace the missing value with another value. For example, with a feature value of data points most similar to the one with the missing value or a mean value of other sample points. In this work, sample points with missing values are removed.

- *Discretisation.* Sometimes it is beneficial to discretise a continuous feature. By doing so, one would decrease the number of possible values. Discrete values are preferred or even sometimes outright necessary for some of the machine learning algorithms. Selecting optimal binning intervals is a non-trivial task, and multiple approaches should be considered.

- *Data normalisation.* Different features are often measured in different units. Moreover, features can have substantially different scales. Normalisation is used to scale the features and bring them to the same level of magnitude. For some machine learning algorithms, normalisation is necessary to achieve meaningful results. In this work, standardisation was used.

- *Feature selection.* Datasets often contain features that hold no helpful information for predicting specified labels. Redundant features should not be included in the list of input variables to keep the model parsimonious. The dimension of input variables should be kept small due to the *curse of dimensionality* [8]. Often the features are chosen iteratively after training multiple possible models or using regularisation[9].

- *Feature construction.* New features can be created by mapping a function on the original data and transforming them.

2.1.2 Algorithm Selection

When the dataset is prepared, a suitable machine learning algorithm is selected. There is no single model that would be best for all situations. This notion is typically referred to

as "No Free Lunch Theorem" of machine learning [10]. Usually, algorithm selection is an iterative process. After evaluating the initial model, the process goes back to algorithm selection, and another machine learning algorithm is selected, as depicted in Figure 1.

Each iteration, the model is adjusted, or another algorithm is selected based on objective statistical measures. It is impossible to know beforehand which model is the most suitable; however, some properties of the analysed data can guide us towards the right choice. The complexity of the relation between features and labels decides whether it is better to use more flexible models or opt for more simple and interpretable models. In Figure 2 is shown how an increase in model flexibility is generally associated with a decrease in interpretability.

Some algorithms take a longer time to train and optimise, so it may be not good to use them if time is of concern. Knowledge of prior model performance on a particular problem, similar in nature, can help choose the suitable algorithm. For example, convolutional neural networks perform well on image recognition [11].

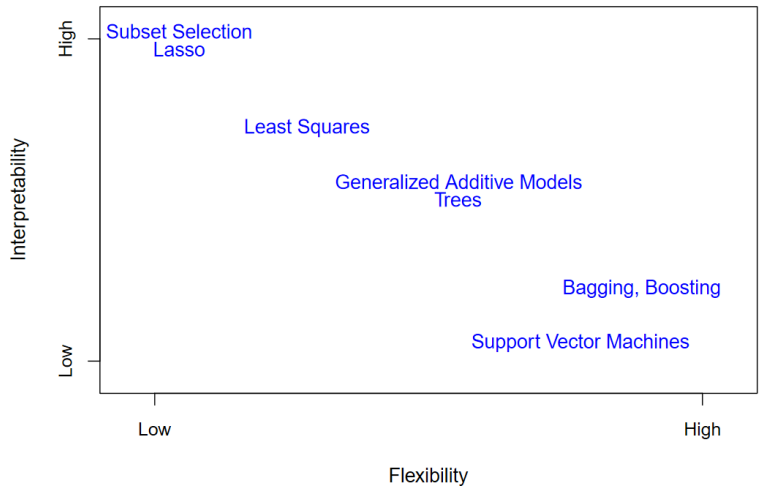


Figure 2: Trade-off between model flexibility and model interpretability, adopted from [12].

2.2 Support Vector Machine

Support vector machine is a method used for classification popularised by [13]. It is also one of the methods used in this thesis. This section presents the algorithm, as described in [9] and [12]. At first, *maximal margin classifier* is described. Maximal margin classifier

is then adjusted to define support vector classifier, and finally, support vector machines are defined. For simplicity, binary classifiers are defined. At the end of the section, the classifiers are generalised for multiclass classification problems.

2.2.1 Maximal Margin Classifier

The idea is to find in a hyperplane in the space of features that would separate the instances of the sample set into two distinct classes. All the instances of one class are on one side of the hyperplane, and the rest are on the other side.

Hyperplane in k-dimensional space is an affine subspace of dimension $k-1$ and can be defined as:

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k = 0. \tag{1}$$

If vector of variables $X = (x_1, x_2, \ldots, x_k)^\top$ satisfies (1), it lies on the hyperplane. Alternatively, vectors can lie in one of the halves, depending on whether

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k > 0, \tag{2}$$

or

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k < 0.. \tag{3}$$

The task is to find $\beta = (\beta_0, \beta_1, \beta_2, ..., \beta_k)^\top$ under condition that all the feature values of instances of one class satisfy (2), while others satisfy (3).

One such hyperplane is shown in Figure 3. The hyperplane acts as a decision boundary, and new instances can be classified depending on which side of the decision boundary they are. Notice that by rotating and shifting the hyperplane by a small amount, it still perfectly separates the sample points by their classes.

Maximal margin classifier deals with finding optimal separating hyperplane out of infinitely many available. When a new observation is classified, the further it lies from the decision boundary, the more confident we are that the observation is classified correctly. This idea is analogically used in selecting the optimal hyperplane. For each candidate hyperplane, perpendicular distances to the sample points are calculated. The smallest of those distances is called a margin, and the observations closest to the hyperplane, that define the margin are called *support vectors*. Maximal margin classifier is learned by

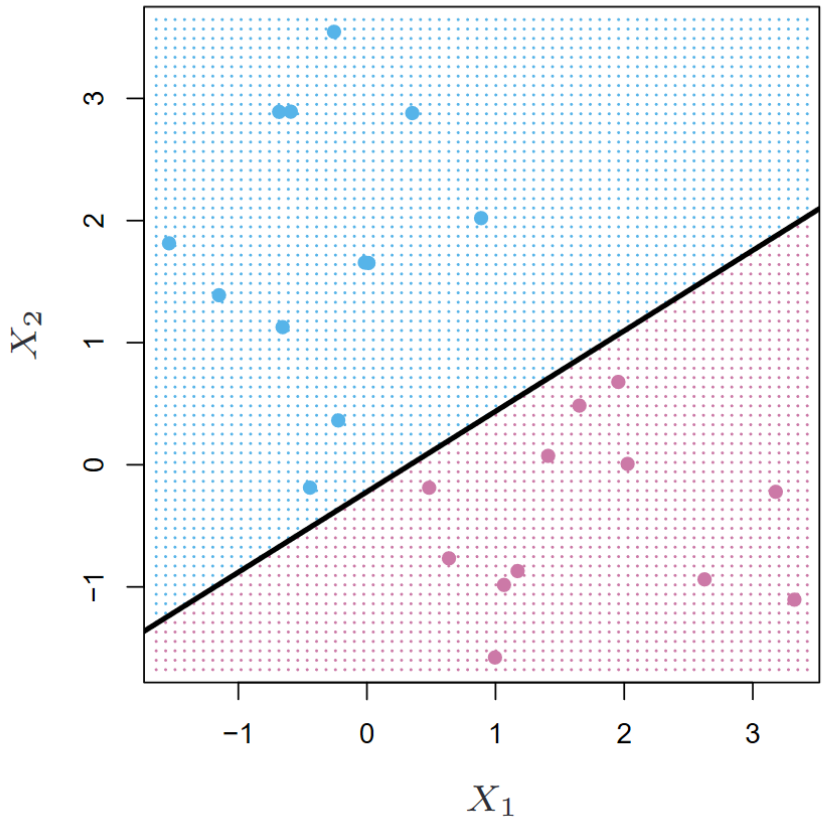


Figure 3: An example of hyperplane splitting a 2-dimensional space into two halves. All the instances of blue class fall above the separating hyperplane. In the same way, all the instances of red class fall below the separating hyperplane. Adopted from [12].

algorithm, which chooses the hyperplane with the largest margin, given a labelled training sample set of features. The existence of a separating hyperplane is conditioned on the linear separability of the dataset.

2.2.2 Support Vector Classifier

If a hyperplane cannot perfectly separate the classes, the conditions need to be eased to allow for misclassification of some sample points. The algorithm still seeks to maximise margin, but the conditions of optimisation problems are adjusted.

Some data points are allowed to be closer to the hyperplane than the margin or even fall on the other side of the hyperplane than the decision boundary dictates. This is represented by a slack variable $\xi = (\xi_1, \xi_2, \ldots, \xi_n)$, a distance proportional to margin

width by which sample points are on the wrong side of the margin line.

All the values located further from the separating hyperplane than is the margin width, and at the same time are on the correct side of the hyperplane have $\xi_i = 0$. All the sample points on the wrong side of the margin have $\xi_i > 0$. As the value of ξ_i is proportional to the width of the margin, sample points located on the wrong side of the hyperplane have $\xi_i > 1$.

The total amount by which datapoints breach the limit set by the margin is constrained $\sum_{i=0}^{n} \xi_i < constant$. When expressed in the optimisation problem, the constant is controlled by a parameter C. Smaller values of parameter C correspond to a more soft margin that gives the algorithm more freedom to ignore some sample points. Large values of the parameter make the algorithm more strict in this regard.

2.2.3 Kernel Trick

So far, the methods of finding a separating hyperplane have offered us only a possibility to construct a linear decision boundary. In many cases, the true boundary between the classes is not of linear character, which calls for an algorithm producing nonlinear decision boundaries.

Support vector machine (SVM) handles this problem by introducing a *kernel trick*. The original features are transformed by a chosen function $h(x)$. This increases the dimension of feature space.

In the new feature space, a separating hyperplane is chosen as before. The decision boundary is again linear in the newly transformed feature space. However, in the space of original features, the decision boundary is now nonlinear.

Due to curse of dimensionality selecting complex functions which vastly increase dimension of feature space can be time consuming. Support vector machines algorithm handles the problem well. When finding the hyperplane, the optimisation problem is expressed in dual form. As stated in [9], the *Lagrange dual function*

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \langle h(x_i), h(x_j) \rangle, \tag{4}$$

and the solution function

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i \langle h(x), h(x_i) \rangle + \beta_0, \tag{5}$$

where i, j are indexes of observations, $y \in \{-1, 1\}$ are the labels with classes encoded, and α_i are Lagrange multipliers, both contain transformed features $h(x_i)$ only through inner product. Now the sample point features need not to be transformed when training, if the right *kernel function* is used. *Kernel function*

$$K(x, x') = \langle h(x), h(x') \rangle, \tag{6}$$

computes the inner product in the space of transformed features. This method, also known as *kernel trick,* vastly speeds up the process of learning the classifier and allows for handling of high dimensional feature space. The various kernel functions and their advantages can be found for example in [14].

The choice of kernel function is an essential task. In theory, we should choose a function, which makes the sample set mostly linearly separable in the new feature space. Generally, increasing the complexity of the kernel function makes the transformed dataset linearly separable. It is ill-advised to use too complex kernel functions [15]. Despite the kernel trick, the curse of dimensionality is still applicable. In addition to prolonging the training time needed, using complex kernel functions can result in *overfitting*.

2.2.4 Generalisation for Multi Class Problem

The hyperplane splits the feature space into two. As such, it is trained to distinguish between two classes. If the labels contain multiple classes, the procedure needs to be extended. There are two common approaches.

The *one-vs-one* approach trains an SVM model for each distinct pair. Predictions are based on the votes of the trained models.

The *one-vs-all* approach also trains multiple SVM models. One class is selected, and a separating hyperplane is found that separates it from the rest of the classes. The procedure is repeated for each class, resulting in one SVM model per class. New observations are classified according to which model is the most "certain" of the classification. The level of certainty is represented by the distance of the data point to the separating hyperplanes. In this work we use one-vs-all approach.

2.3 Overfitting

Citing [16]:

> "A model overfits the training data when it describes features that arise from noise or variance in the data, rather than the underlying distribution from which the data were drawn. Overfitting usually leads to loss of accuracy on out-of-sample data."

Classification models are used to capture the relationship between the features of observations and their class membership. Using more flexible models could also capture the randomness that comes with each dataset. As a result, we should only include the necessary number of predictors and keep the model simple [17].

Classic example of overfitting is fitting a polynomial of order $n-1$ to n points. Suppose the underlying relationship between the dependent and independent variables is linear. The n observations are results of the true data generating function in combination with random deviations. Using a polynomial of degree $n-1$ in regression analysis, would pass exactly through the n training data points yielding zero error. However, such model hardly reflects the true linear relationship. In this case, using linear regression would be the better option. Restricting the set of all possible model functions to only linear functions makes the model less flexible and results in higher training error.

2.3.1 Train-Test Split

When evaluating the performance of classifiers, calculating metrics on the sample set that was used for model training can be misleading. Overfitting loses on the accuracy of new out-of-sample observation classification, but it fits the training data well.

It is good to split the dataset randomly into a training set and a test set to simulate out-of-sample data. The model is then trained on the training set. The test set is used for model validation.

2.3.2 Cross-Validation

When there is not enough data, the validation score would be highly affected by the randomness of the train-test split. The starting dataset can be divided into k number of

folds. The algorithm is then scored so that $k-1$ folds are used for training, and the one fold left out of the training process is used for validation. This process is repeated for all of the folds, each time leaving one for model validation, whilst training the model on the rest of the folds. After k iterations, the resulting scores are averaged, and the test accuracy is approximated.

2.3.3 Grid-Search

There is not much theory that would help with the right choice of the parameter C or of the kernel function of SVM, mentioned in 2.2.2 and 2.2.3 respectively. These values are chosen by a trial and error process. Not only SVM has such parameters that need to be set. Many other models rely on input that needs to be set prior to the training part. Such inputs are called *hyperparameters*.

A semi-automated process that helps with the choice of hyperparameters is *grid-search*. At first, possible values of hyperparameters are set manually. After that, all possible combinations of hyperparameters are used to train the models. The models are validated with a scoring method of choice. The best performing model is selected, and the combination of hyperparameters of that model is pronounced the most suitable.

Grid-search is accompanied by cross-validation to prevent overfitting. For each combination of hyperparameters, a model is trained k times depending on the number of cross-validation folds. It is time demanding but useful to avoid overfitting.

2.4 Attractor Reconstruction Analysis

The attractor reconstruction method is a novel method for analysing blood pressure data. Its effectiveness in combination with machine learning was demonstrated in [18]. In this chapter we also used [19] and [20], where the theory is described in greater detail.

A heart of a rat is a complex nonlinear system. To describe the system from a single chaotic time series data, sophisticated methods must be used. This method tries to reconstruct the phase space of the system. More specifically, from one dimensional observations of a signal, an object in a reconstruction space of larger dimensions is created that has similar properties (is diffeomorphic) as the original phase space dynamics [21][22]. In [23] is argued, that this can indeed be achieved, if the dimension of the reconstruction space

is large enough. The original dynamic is a strange attractor defined in [24].

2.4.1 Reconstructing the Attractor

First, the time series is represented in a three dimensional phase space. Takens' embedding theorem [23] is used. Starting with a time series x(t), we define

$$y(t) = x(t - \tau), \qquad z(t) = x(t - 2\tau).$$

This way, the time series data can be plotted in an (x,y,z) reconstruction space, as in Figure 4. From now on, the reconstructed dynamics and their transformations will be referred to as atrractors for simplicity sake.

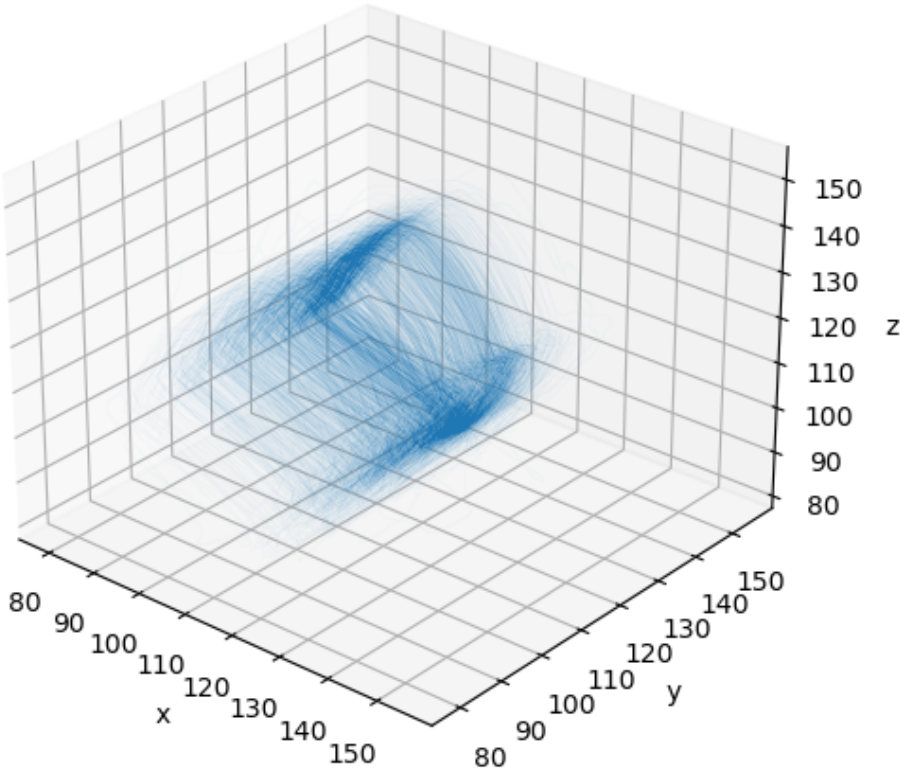


Figure 4: An arterial blood pressure signal of one examplary rat plotted in an (x,y,z) phase space.

2.4.2 Removing Baseline Variation

An animal does many activities throughout the day, some are active, and some are passive. The blood pressure responds accordingly to the constant change in the activity of the animal and the level of stress in the animal. As a result, the time series of blood pressure are non-stationary. The method is insensitive to the average height of the signal but instead uses information about the functional profile. Shifting the time series by a constant $x(t) \to x(t) + c$ would cause a similar shift in the variables y and z. In the three-dimensional phase space, this would result in a shift of the attractor in the direction of the vector $(1, 1, 1)$.

$$(x(t), y(t), z(t)) \to (x(t) + c, y(t) + c, z(t) + c)$$

To remove the baseline variation, the reconstructed attractor is projected onto a plane orthogonal to the vector $(1, 1, 1)$. New variables are defined. We decided to define the variable u differently to the [19] and [20]. Whilst still orthogonal to the variables w and v, it now also represents the numerical second derivative of the time series.

$$w = \frac{1}{3}(x + y + z), u = \frac{1}{\sqrt{6}}(x - 2y + z), v = \frac{1}{\sqrt{2}}(x - z).$$

Shifting the time series by a constant affects only the variable w. The attractor can now be projected onto the (u,v) plane, as in Figure 5. The resulting shape should only be affected by the changes to the shape of the waveform.

2.4.3 Constructing a Density

A density on a uniform grid is constructed to compare projected attractors better. The bin width is specified, and a two-dimensional histogram is created. The volume is then normalised to be one.

2.4.4 Choosing the Right τ

While using an arbitrary value of time delay τ recreates the attractor [23], some choices of τ perform better in some cases. Various methods are discussed and proposed in [25] and [26]. Many methods use mutual information or autocorrelation to choose τ, but in this work was used the shape of the reconstructed attractor. The choice was inspired

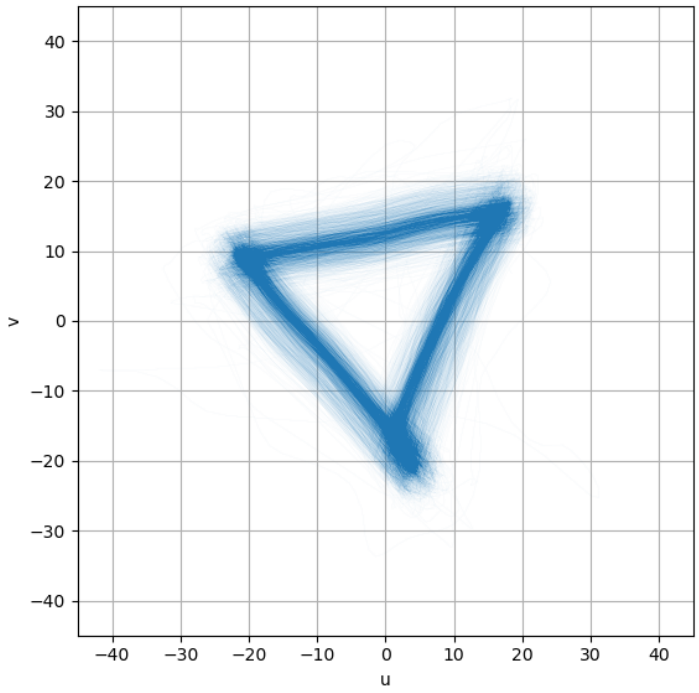


Figure 5: The exemplary signal plotted in a (u,v) space.

by [19] and [20]. Suppose the time series is periodic with period T, then choice of the τ parameter to be $T/3$ or $2T/3$ results in a reconstructed attractor with a threefold rotational symmetry about the origin [27]. Arterial blood pressure and many other time series samples are close to, but not perfectly periodic. The value of τ is chosen by finding the most symmetrical attractor. Let $D_1(\tau)$ denote the constructed density on a uniform grid. A density $D_2(\tau)$ from the attractor rotated by $\frac{2\pi}{3}$ is created, as well as $D_3(\tau)$ density created by rotating the attractor by $\frac{4\pi}{3}$. Combining three rotated histograms, new density is defined as

$$D_s(\tau) = \frac{1}{3}(D_1(\tau) + D_2(\tau) + D_3(\tau)).$$

Note that the D_s density has a three-fold symmetry from the construction. This can be seen in Figure 6

If the attractor has a three-fold symmetry, then $D_s = D_1$. Finding the optimal τ is now a problem of finding the value of τ for which the distance between $D_1(\tau)$ and $D_s(\tau)$ is minimal. Let

$$S(D_1(\tau)) = \|D_1(\tau) - D_s(\tau)\|_2,$$

be the symmetry measure of the density, the optimal τ is set by finding the minimum



Figure 6: Left: Histogram $D_1(\tau)$ constructed from the exemplary signal projected onto the (u,v) phase space. We can see a higher density at the vertices. Right: Histogram $D_s(\tau)$. Symmetrical from the construction. Sum of the rotations of the exemplary $D_1(\tau)$ histogram.

of the $S(D_1(\tau))$ function. Recall that the attractor has a three-fold rotational symmetry about the origin for $\tau = \frac{T}{3}$ and $\tau = \frac{2T}{3}$, which is illustrated in Figure 7.

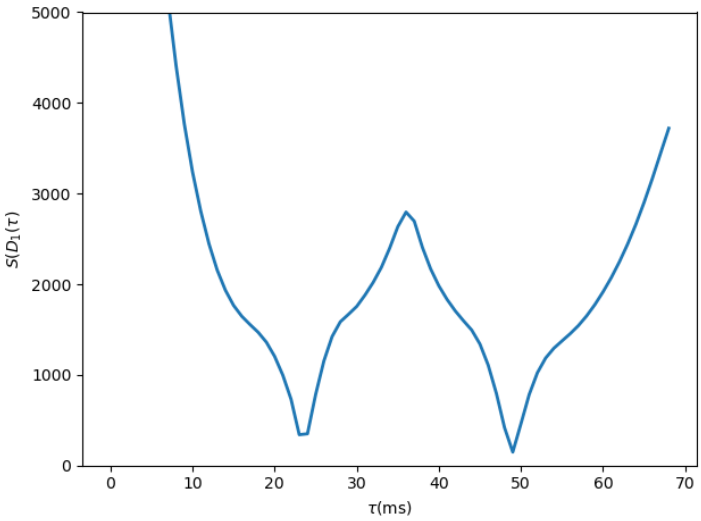


Figure 7: Plot of the symmetry measure function. Two local minima can be observed. The first minimum should correspond to the value of $\tau = \frac{\text{Heart beat period}}{3}$. The second minimum should correspond to the value of $\tau = \frac{2\times\text{Heart beat period}}{3}$

3 Data

The methods described in section 2 were employed on data of rats. The rats were located in a lab with a light cycle of 12 hours of light and 12 hours of dark. An automatic measuring device recorded multiple types of information (locomotion activity, pulse pressure, etc.) at a high recording frequency (telemetry). The rats were implanted with a small chip. Three or four times per hour, a 5-minute record of the rat's blood pressure was sent wirelessly from the chip. The records were taken at a rate of 500Hz. All data were collected by Ľuboš Molčan et al. at the Faculty of Natural Sciences, Comenius University, Bratislava, in the experiments aimed at understanding circadian rhythms in mammals. We worked with datasets with three different levels of granularity.

3.1 Aggregate Data

Table 1: Descriptive statistics of the first dataset.

	HanSD	*WT*	*TGR*	*SHR*
Count of rats	15	75	29	72
Recordings per hour	3	4	3	4
Count of recordings	1080	7200	2088	6912
Measurements per one recording	1	1	1	1

The first dataset contains data of all the rats, i.e. 15 *HanSD*, 72 *SHR*, 29 *TGR*, and 75 *WT*, as is also presented in Table 1. Each rat is represented with tens of the 5-minute records taken in a span of 24 hours. One 5-minute record is one sample point with multiple statistical characteristics as features. As an example, sample points of one rat plotted in time are illustrated in Figure 8. The characteristics are:

- Mean blood pressure, which is the average blood pressure throughout the whole segment.
- Systolic blood pressure, which is the average of the peaks.

- Diastolic blood pressure, which is the average of the negative peaks.
- Pulse pressure, the average distance between the systolic and diastolic blood pressure.
- Heart rate, expressed in beats per minute.
- Activity, representing the number of times the motion sensor recognised a movement of the rat.

Each row of the dataset also includes additional information:

- Timestamp.
- Id of the rat.
- Light intensity, a value of either 150 or 0 representing whether the record was taken during light or dark phase respectively.

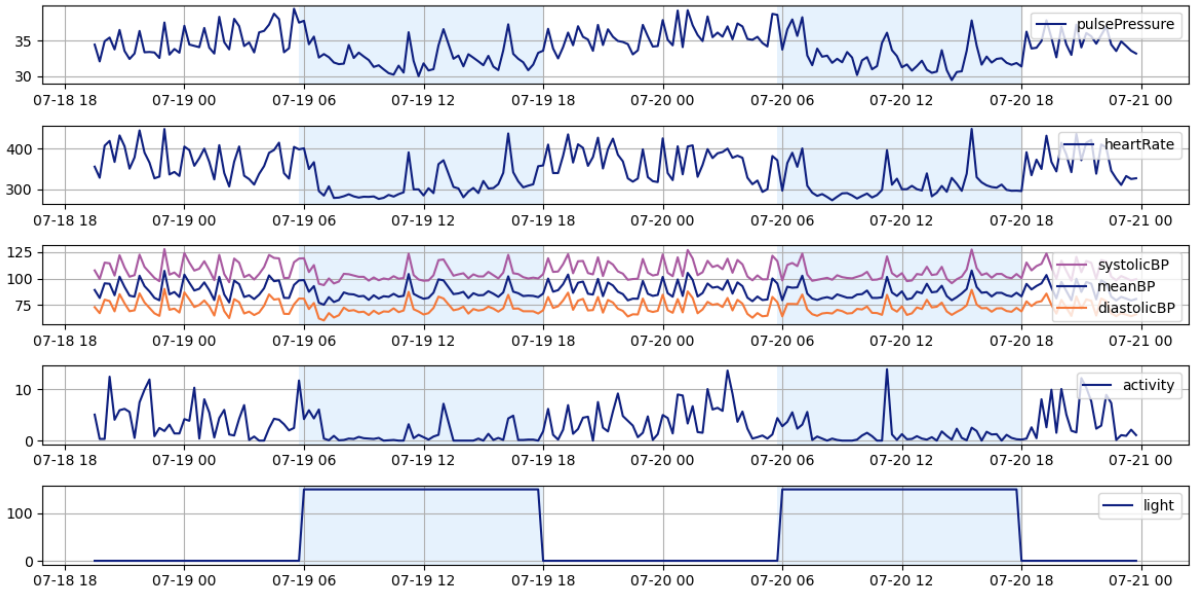


Figure 8: The time series data of the first form. The data of one rat plotted against time. The blue regions represent time periods during which an artificial lights were turned on.

Table 2: Descriptive statistics of the second dataset.

	HanSD	*WT*	*TGR*	*SHR*
Count of rats	7	6	5	7
Recordings per hour	3	4	3	4
Count of recordings	504	576	360	672
Measurements per one recording	≈ 2000	≈ 2000	≈ 2000	≈ 2000

3.2 Beat-to-Beat Data

Next, a dataset of Beat-to-beat data is analysed. It again consists of 5-minute recordings taken during 24-hour intervals. This time the data are more granular than the data in the previous dataset. Each 5-minute recording is one file containing roughly 2000 rows. The descriptive statistics of the dataset are included in Table 2.

Each row represents a single heart contraction. A timestamp of the heart contraction is coupled with the corresponding systolic blood pressure measurement (the blood pressure at the time of heart contraction), and with the length of the interbeat interval (the time between the last and current heartbeat). Data of one exemplary recording is shown in Figure 9.

The granularity of this dataset is approximately 0.15 second, i.e. roughly the average time between heartbeats of the rats. Due to the increased memory and space demands, the dataset contains only a reduced number of rats. We used 7 *HanSD*, 7 *SHR*, 5 *TGR*, and 6 *WT* rats.

3.3 Waveform Data

The third dataset describes the original blood pressure signal data. As in the second dataset, this dataset consists of the 5-minute recordings, taken in the course of 24 hours, saved in the form of separate files. Each file has ca. 150000 measurements of blood pressure. The measurements represent blood pressure signals with a sampling rate of 500Hz. Only data of one rat are selected per strain.

Plotted in Figure 10 is a six seconds recording of arterial blood pressure from the data.

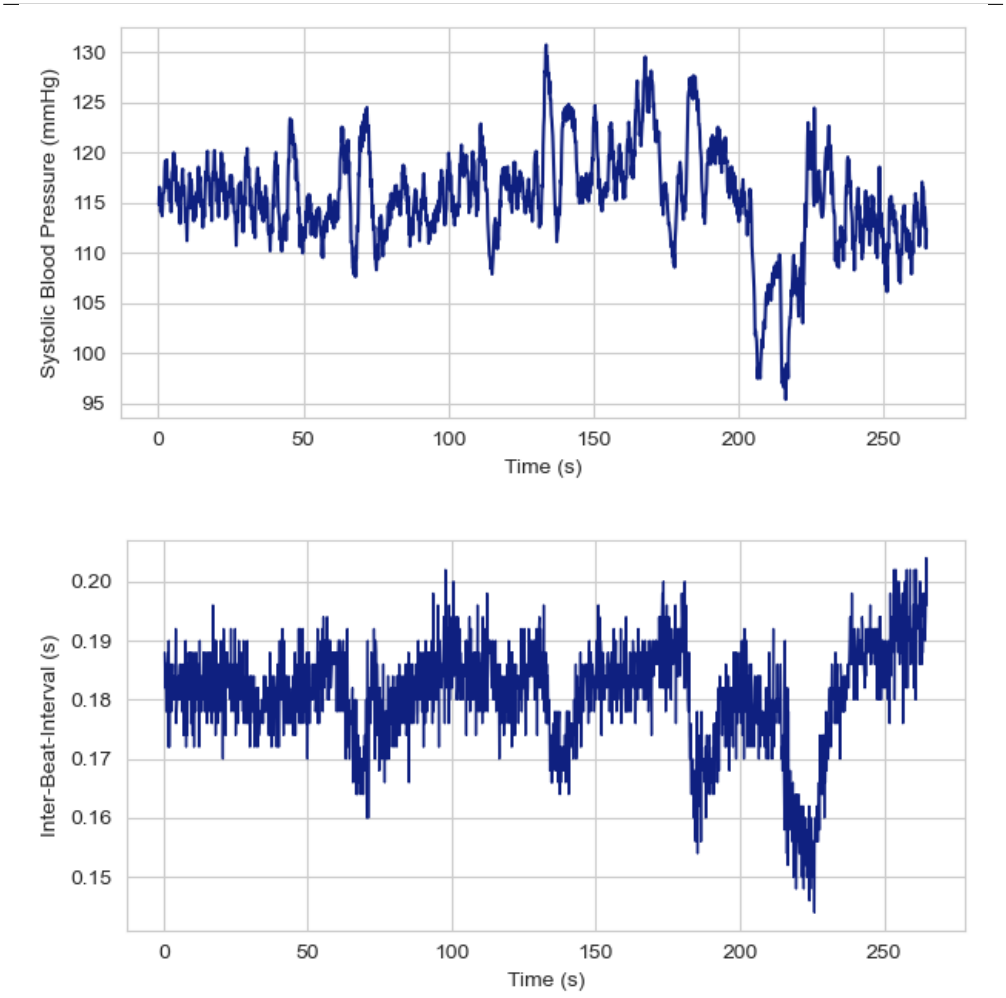


Figure 9: Plots showing one record from the second dataset. One record is represented by a 5-minute time series of systolic blood pressure and by a time series of interbeat interval lengths.

The waveform is changing with every heartbeat. Both systolic and diastolic pressures are increasing and decreasing over time without any observable pattern. The underlying system that creates this signal, combined with the measurement error result in chaotic signal.

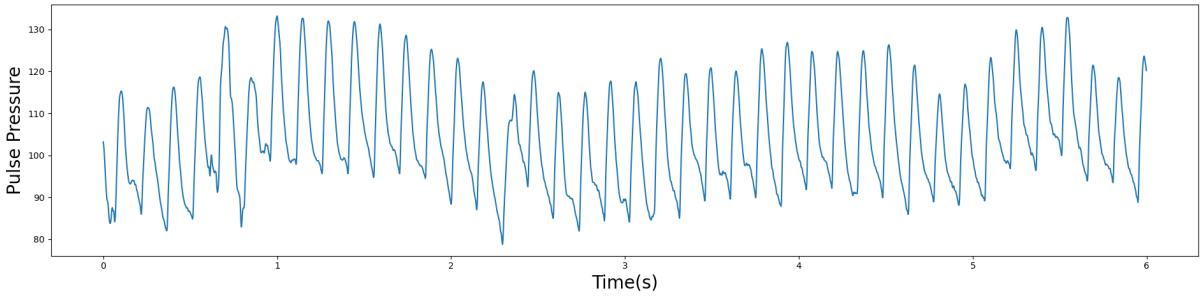


Figure 10: A segment of the time series data of the third form. The data of one rat plotted against time.

Table 3: Descriptive statistics of the third dataset.

	HanSD	*WT*	*TGR*	*SHR*
Count of rats	1	1	1	1
Recordings per hour	3	4	3	4
Count of recordings	72	96	72	96
Measurements per one recording	≈ 150000	≈ 150000	≈ 150000	≈ 150000

4 Applications

4.1 Classification Using Dark/Light Phase Averages

To represent each rat by a small number of statistics, we created averages of the covariates. Rats are active at night, which is illustrated in Figure 8. In the blue regions, which represent the light phase, the time series of activity, heart rate, and pulse pressure are visibly lower. Averages were thus created separately for the light phase and the dark phase. Features used as input variables in the machine learning model were:

- Mean blood pressure - Light phase average.
- Mean blood pressure - Dark phase average.
- Systolic blood pressure - Light phase average.
- Systolic blood pressure - Dark phase average.
- Diastolic blood pressure - Light phase average.
- Diastolic blood pressure - Dark phase average.
- Pulse pressure - Light phase average.
- Pulse pressure - Dark phase average.
- Heart rate - Light phase average.
- Heart rate - Dark phase average.
- Activity - Light phase average.
- Activity - Dark phase average.

As shown in Figure 11, differentiating between hypertensive and normotensive rats should not be difficult, seeing that the rats are divided into two clusters relative to their blood pressure values. Furthermore, when comparing their blood pressure during the light and dark phase and comparing their overall heart rate, the hypertensive *SHR* rats are distinguishable from the hypertensive *TGR* rats. Visually inspecting the scatter plots,

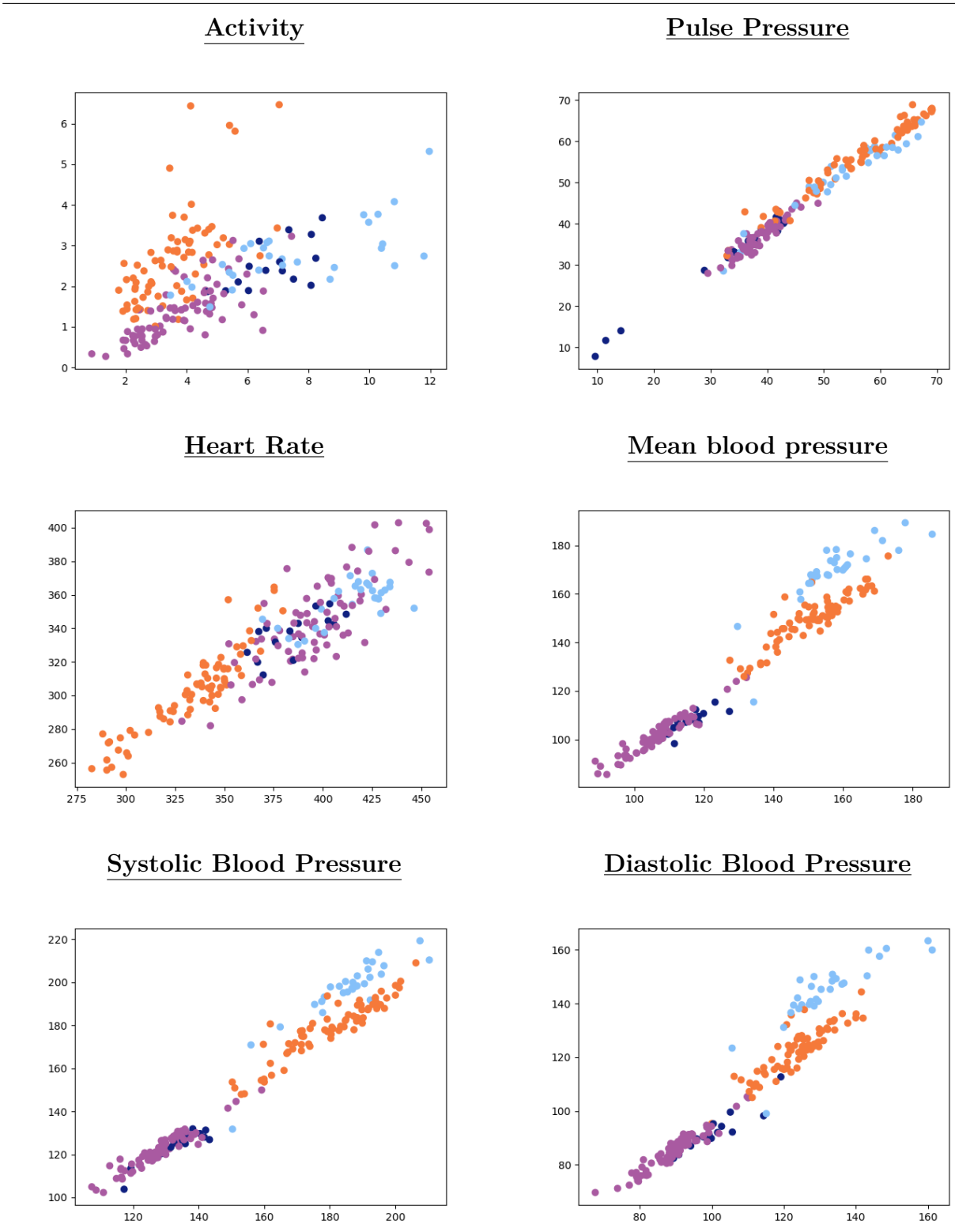


Figure 11: Scatter plots showing the average values of the features during a dark phase (x-axis) and a light phase (y-axis). Each graph shows one feature. One dot represents one rat. The dots are coloured according to the strains of rats. Orange (SHR) and sky-blue (TGR) rats are hypertensive, and purple (WT) and dark-blue (HanSD) rats are normotensive.

it is difficult to discern any pattern differentiating between normotensive *HanSD* and normotensive *WT* rats.

The dataset was divided into a training set, on which the classifiers were fitted, and a test set used to assess the performance of the classifiers. The training set consisted of 70% of the data selected randomly. For testing, we set aside the remaining 30%.

First, the data were normalised by removing the mean and scaling to unit variance. Then an SVM model with a linear kernel was trained on the training set. To choose the value of hyperparameter C, the penalty for misclassifying a datapoint, grid search was used. It considered all the values of C from the set

$$S = \{2^{2i-1} \mid i \in \mathbb{Z}, -2 \leq i \leq 9\}.$$

The performance of the model was evaluated on the test set. Predictions are summarised in a confusion matrix in Table 4. The matrix elements are the counts of instances as predicted by the algorithm. Note that the sum of the first row is six. From that can be deducted that there were six instances of *HanSD* rats in the test set. Of those six, four were classified correctly as *HanSD* strain, and two were classified incorrectly as *WT* strain. Similarly, the algorithm predicted 21 rats of the training dataset to be *WT* rats, yet only 19 of those were actually *WT* rats. The hypertensive rats were all classified correctly.

A classification report of the model is displayed in Table 5. *Precision* is the number of correctly predicted instances of the class divided by the total number of instances classified as that strain. In the confusion matrix, this is the number on the diagonal divided by the sum of the corresponding column. *WT* strain has a precision of 0.90 because of 21 *WT* rats predicted, 19 were actually *WT* rats.

While the model was exact in predicting *HanSD* rats, it has a very low recall. A *recall* is the percentage of the rats with their strain correctly predicted. In the confusion matrix, this is the number on the diagonal divided by the sum of the corresponding row. The model correctly recognised only $\frac{2}{3}$ of the *HanSD* rats from the test set.

The *F1-score* is calculated as

$$2 * \frac{Precision \times Recall}{Precision + Recall}.$$

It is a metric that combines both precision and recall, giving them equal importance.

Lastly, *support* denotes how many instances of the corresponding strain were present in the test set.

Table 4: Confusion matrix of test set classified by the first method.

		Predicted			
		HanSD	*WT*	*TGR*	*SHR*
Actual	*HanSD*	4	2	0	0
	WT	0	19	0	0
	TGR	0	0	6	0
	SHR	0	0	0	27

Table 5: Classification report. The performance metrics of the first method.

	Normotensive		Hypertensive	
	HanSD	*WT*	*TGR*	*SHR*
Precision	1.00	0.90	1.00	1.00
Recall	0.67	1.00	1.00	1.00
F1-score	0.80	0.95	1.00	1.00
Support	6	19	6	27

Overall, the model was successful in classifying most of the rats. It failed to classify some of the *HanSD* rats and predicted them to be of *WT* strain. The advantages of this method are simplicity and easy biological inference. However, this model requires 24 hours of data collection to classify a new rat. In the next section, we inspect a machine learning model trained on more granular data to examine if the process can be quickened.

4.2 Classification Using Beat-to-Beat Data

The following method only uses a 5-minute record to classify the rat strain. The aim is to decrease the time needed to classify a new rat. This time, the moments of five-minute beat-to-beat data were used as features. For both systolic blood pressure and interbeat interval, average values were calculated, as well as standard deviations, medians and quantiles. These values were used as the input variables of the machine learning model.

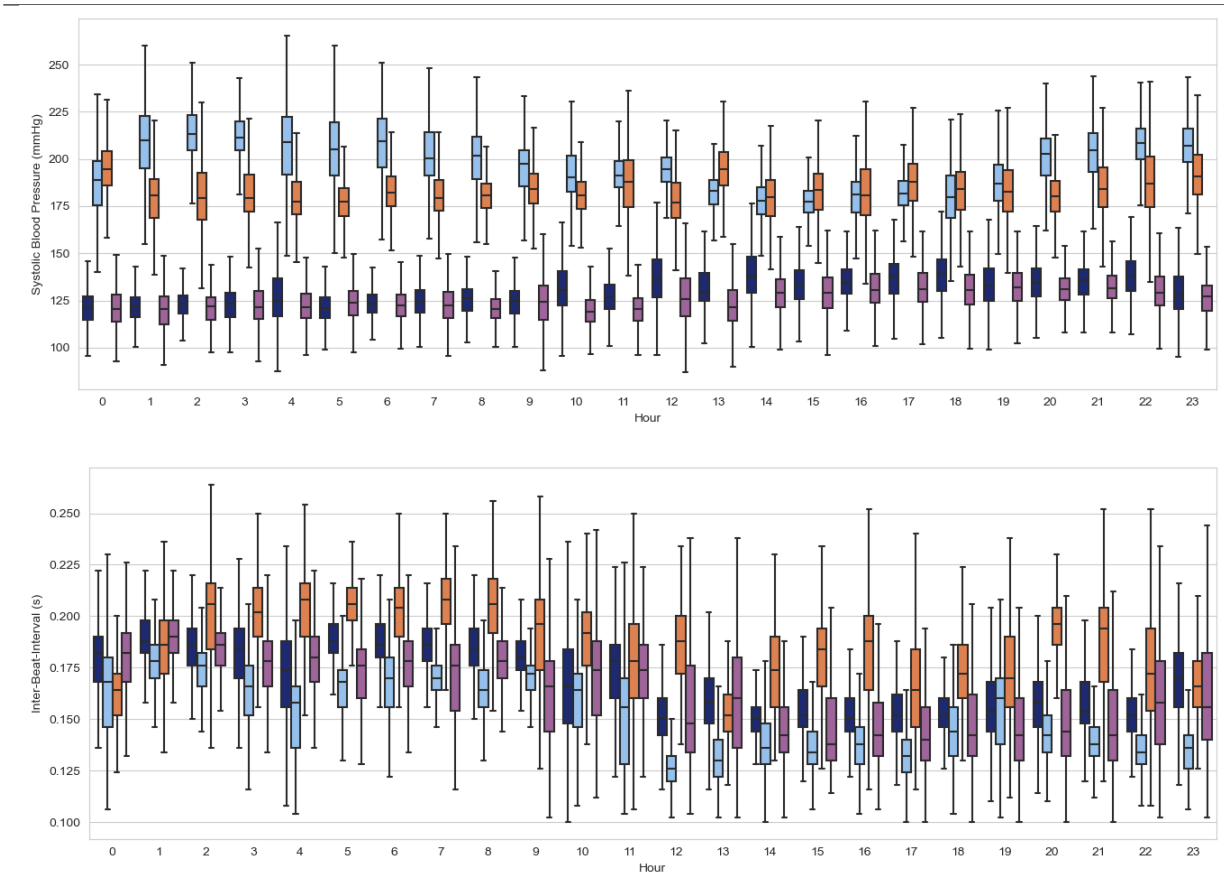


Figure 12: Box-plots of systolic blood pressure and interbeat intervals, ordered in time. Each color portrays different rat genome. From the firs plot we see the difference between the two normotensive strains and the two hypertensive strains. The box-plots of the same strain are not unambiguously correlated in time. Orange (SHR) and sky-blue (TGR) rats are hypertensive, and purple (WT) and dark-blue (HanSD) rats are normotensive.

Analysing box-plots in Figure 12, it can be assumed that the records are not correlated in time. As a result, each record is used as one observation in the machine learning process.

Same as in the previous method, training and test sets were created, the data were scaled, and the SVM model was trained using grid search.

Table 6: Confusion matrix of test set classified by the second method.

		Predicted			
		HanSD	*WT*	*TGR*	*SHR*
Actual	*HanSD*	98	49	2	0
	WT	40	149	0	0
	TGR	1	0	79	19
	SHR	0	0	18	179

Table 7: Classification report. The performance metrics of the second method.

	Normotensive		Hypertensive	
	HanSD	*WT*	*TGR*	*SHR*
Precision	0.71	0.75	0.80	0.90
Recall	0.66	0.79	0.80	0.91
F1-score	0.68	0.77	0.80	0.91
Support	149	189	99	197

The results are included in Table 6. The trained model was used to predict the classes of the test set observations. Despite correctly predicting most of the classes, the model lacks in performance compared to the previous model. Nevertheless, the model used data collected in a much shorter time and could be used to discern between hypertensive and normotensive rats quickly.

4.3 Classification Using Attractor Method

This method tries to capture differences in the arterial blood pressure waveform of the rats. The original raw form of the telemetry data was used. The dataset contains only the arterial blood pressure values and their timestamps.

The data needed to be pre-processed. The dataset contained many faulty data measurements in the form of outliers. To detect outliers, means and standard deviations were calculated for each 5-minute segment separately. If the measurement was more than three times the standard deviation from the mean, it was marked an outlier. This outlier detection method was chosen by trial after visually inspecting that it performed well. The outliers were easily distinguishable, and as such, no different complex methods were needed to be used. The detected outliers were ignored when creating attractor.

One 5-minute segment was used to reconstruct one optimal attractor, as mentioned in subsection 2.4. For each segment, the symmetry measure function $S(D_1(\tau))$ from subsection 6 was calculated. To estimate $S(D_1(\tau))$ function, for $t = 1, 2, \ldots, 100$, the following steps were repeated.

1. The original signal was delayed by t value specified. The first lagged values were created.

2. The shifted signal was delayed by t again. The second lagged values were created.

3. The data were transformed; The u and v coordinates were created, as is described in subsection 2.4.2.

4. The 2D histogram was created.

5. The data were rotated by $\frac{2\pi}{3}$ with a rotation matrix, and the second histogram was created.

6. The data were rotated by another $\frac{2\pi}{3}$ and the third histogram was created.

7. The histograms were combined, and the symmetry measure was calculated, as is described in subsection 2.4.4.

To recreate the optimal attractor, the minimum of the symmetry function was found. Thus was found the optimal t_{opt}. Because the data were sampled 500 times per second, we find the optimal $\tau_{opt} = 0.002 t_{opt}$.

The most symmetrical attractor can now be recreated for each of the 5-minute segments. From the attractors, the features can be extracted.

Increase in the amplitude of the original signal results in an increase in the attractors' size. Moreover, the clockwise rotation of the attractor arises from the increased concavity of down-strokes of the waveform. On the other hand, increased convexity of upstroke in blood pressure waveform causes the attractor not to have a uniform density along the edges. These and more features are further explained and described in [27], along with their biological interpretation and derivation.

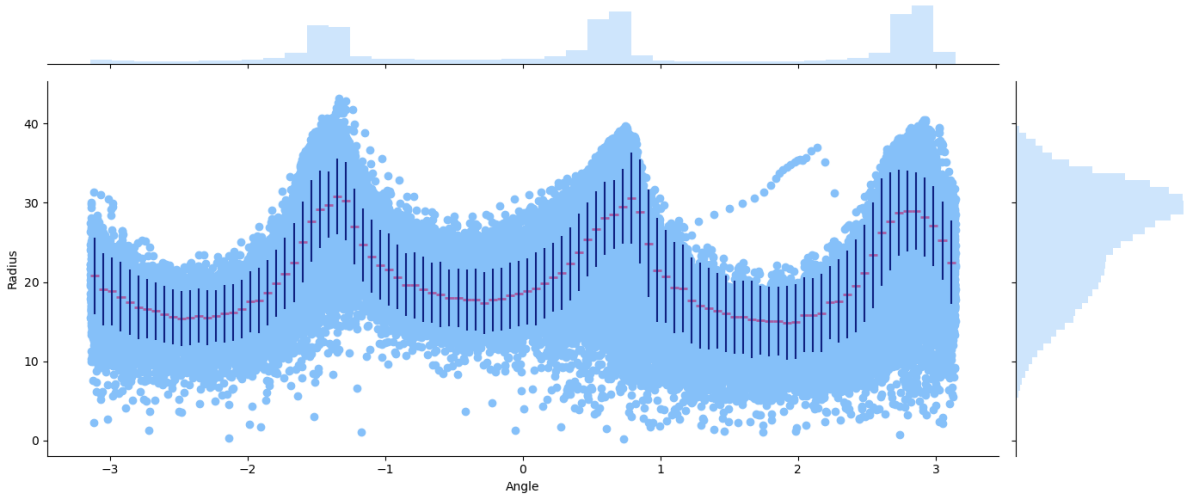


Figure 13: One exemplary optimal attractor plotted in polar coordinates. The angle variable is discretised. For each bin, a mean value is marked with purple colour and the standard deviation with dark-blue colour.

In order to extract these features, the attractor is transformed into polar coordinates. Because in the u,v coordinates the attractor takes the form of a triangle, three local maxima are distinguishable in polar coordinates. The maxima can be seen in Figure 13. Finding the angle of the first local maximum gives us the first feature used in the machine learning process - i.e., the attractor rotation. Moreover, the maximal value can be understood as the circumscribed circle of the attractor, and it is used as the second feature. Another extracted feature describing the size of the attractor is the value at the

minimum. This value represents the inscribed circle.

Because the data points are scattered, the data are discretised along the angle axis to find the mentioned maxima and minima. The bins are created, and for each bin, a mean value of the radius is calculated. The means are then compared to find the maximum and minimum values. Along with the means, standard deviations are calculated for each bin. The deviations in maxima describe the width of the triangles at the vertices. Similarly, deviations at minima describe the width at the edges. The counts of sample points in the maxima and minima are also calculated. The ratio of the two is the last feature describing the extracted attractor.

While reconstructing the attractors, we obtain τ_{opt}, a time delay found by minimising the symmetry function. It contains information about the estimated heart rate and will be used as a feature in the machine learning process.

These features can then be used to train a machine learning classification model. The dataset is split into the training and the test set. After scaling the features, SVM is employed along with grid search. When the best model is trained and selected, it is used to make predictions on the test set.

Table 8: Confusion matrix of test set classified by the third method.

		Predicted			
		HanSD	*WT*	*TGR*	*SHR*
Actual	*HanSD*	19	1	0	0
	WT	1	27	0	0
	TGR	0	0	27	0
	SHR	0	0	0	26

This method was very successful. The classifier was able to classify the rats very precisely. A comparison of selected attractors can be seen in Fig.14. The differences were hard to pinpoint by human eyes, but it was no problem for SVM. The confusion matrix is displayed in Table 8, and the performance metrics are in Table 9. Out of the 101 observations from the test set, only two were misclassified by the model.

Table 9: Classification report. The performance metrics of the third method.

	Normotensive		Hypertensive	
	HanSD	*WT*	*TGR*	*SHR*
Precision	0.95	0.96	1.00	1.00
Recall	0.95	0.96	1.00	1.00
F1-score	0.95	0.96	1.00	1.00
Support	20	28	27	26

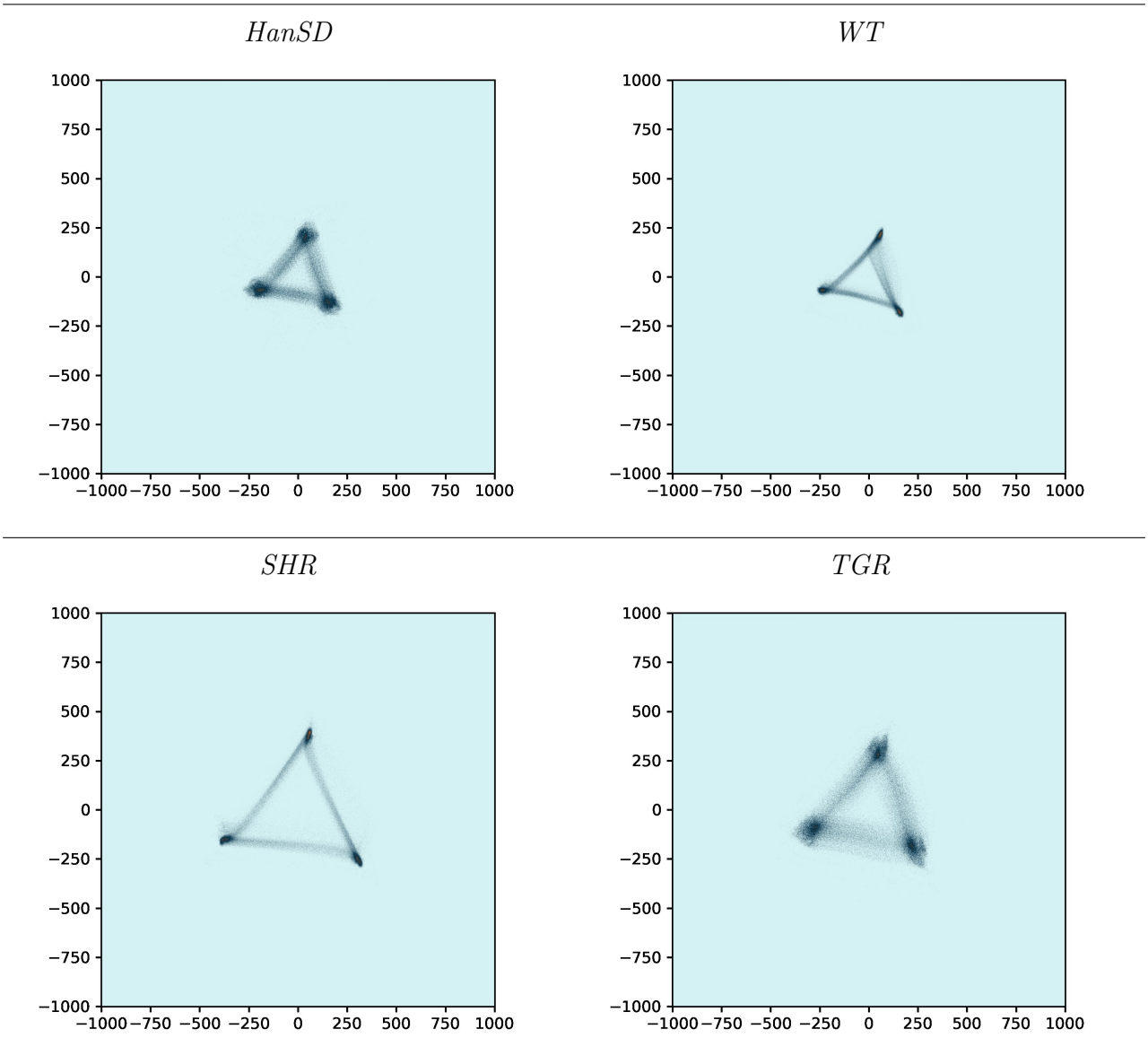


Figure 14: Comparison of the exemplary densities. The attractors in the second row are of hypertensive rats. Higher values of blood pressure results in attractors of larger size.

Conclusion

Three methods were applied to three datasets to train a model that would classify rat strains. The datasets were different in characteristics that they describe and in the granularity of the data. The long term goal is to find a classifier that would classify the rat strains near real-time. We used modern machine learning approaches, combining biological knowledge with the black-box approach so popular in machine learning. The work encompasses a considerable amount of code written in the Python programming language. The practical part of the thesis consisted of hours of handling large datasets, data preparation and proper usage of Python libraries popular for machine learning (e.g., Scikit-learn, SciPy, Multiprocessing, Pandas) and for visualisation (e.g., Seaborn and Matplotlib). We also provide scripts for the novel method of attractor reconstruction, which cannot be found in any free accessible libraries. The scripts include all the needed steps, including data transformation, finding optimal attractor, feature extraction and visualisation. The thesis also explains the theory of machine learning practices and algorithms required to successfully train and validate machine learning models. Finally, the results are analysed, and possible future improvements are drawn up.

The methods were successful in finding the differences between the strains. The first method classified the test observations very accurately. However, it used 24-hour data. The second model was trained on more granular data. This time, only a 5-minute recording of beat-to-beat data was needed to classify a rat strain. The model was only semi-successful. Despite the decrease in accuracy of the classifier, the model has the advantage of the time needed for data gathering. Both models' successes are overshadowed by the accuracy and efficiency of the last model. The attractor reconstruction method demonstrated its usefulness by having the best results while only using a single 5-minute record for classification.

The attractor reconstruction method creates an attractor that reflects the original blood pressure waveform. The question "How long must be the waveform recording to reliably reconstruct the attractor?" is not explored in this thesis and is the natural continuation of the work. The method should also be validated on a larger dataset before being used in research or commercially. If it does not perform as accurately as needed, we would advise adjusting the machine learning model. Algorithms like decision trees,

random forest or k Nearest Neighbours could be tested. Moreover, different features of the attractors can be extracted. Another interesting extension of this work is to classify the reconstructed attractors with image recognition algorithms. The attractor reconstruction method could also be used to pre-process the features for unsupervised learning, hoping to find new clusters or patterns within the strains.

References

1. Meneses, A. & Hong, E. Spontaneously hypertensive rats: a potential model to identify drugs for treatment of learning disorders. *Hypertension* **31,** 968–972 (1998).

2. Lerman, L. O., Chade, A. R., Sica, V. & Napoli, C. Animal models of hypertension: an overview. *Journal of Laboratory and Clinical Medicine* **146,** 160–173 (2005).

3. Lin, H. Y., Lee, Y. T., Chan, Y. W. & Tse, G. Animal models for the study of primary and secondary hypertension in humans. *Biomedical reports* **5,** 653–659 (2016).

4. Okamoto, K. & Aoki, K. Development of a strain of spontaneously hypertensive rats. *Japanese circulation journal* **27,** 282–293 (1963).

5. Kotsiantis, S. B., Zaharakis, I. & Pintelas, P. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering* **160,** 3–24 (2007).

6. Kotsiantis, S. B., Kanellopoulos, D. & Pintelas, P. E. Data preprocessing for supervised leaning. *International Journal of Computer Science* **1,** 111–117 (2006).

7. Hodge, V. & Austin, J. A survey of outlier detection methodologies. *Artificial intelligence review* **22,** 85–126 (2004).

8. Bellman, R. *Adaptive Control Processes: A Guided Tour* ISBN: 9780691079011. `http://www.jstor.org/stable/j.ctt183ph6v` (Princeton University Press, 1961).

9. Hastie, T., Tibshirani, R. & Friedman, J. *The elements of statistical learning: data mining, inference, and prediction* (Springer Science & Business Media, 2009).

10. Shalev-Shwartz, S. & Ben-David, S. *Understanding machine learning: From theory to algorithms* (Cambridge university press, 2014).

11. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *nature* **521,** 436–444 (2015).

12. James, G., Witten, D., Hastie, T. & Tibshirani, R. *An introduction to statistical learning* (Springer, 2013).

13. Vapnik, V. N. *The Nature of Statistical Learning Theory* ISBN: 0387945598 (Springer-Verlag, Berlin, Heidelberg, 1995).

14. Yekkehkhany, B., Safari, A., Homayouni, S. & Hasanlou, M. A comparison study of different kernel functions for SVM-based classification of multi-temporal polarimetry SAR data. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* **40,** 281 (2014).

15. Ben-Hur, A., Ong, C. S., Sonnenburg, S., Schölkopf, B. & Rätsch, G. Support vector machines and kernels for computational biology. *PLoS Comput Biol* **4,** e1000173 (2008).

16. Webb, G. I. in *Encyclopedia of Machine Learning* (eds Sammut, C. & Webb, G. I.) 744–744 (Springer US, Boston, MA, 2010). ISBN: 978-0-387-30164-8. https://doi.org/10.1007/978-0-387-30164-8_623.

17. Hawkins, D. M. The problem of overfitting. *Journal of chemical information and computer sciences* **44,** 1–12 (2004).

18. Lyle, J. V. *et al. Beyond HRV: Analysis of ECG signals using attractor reconstruction* in *2017 Computing in Cardiology (CinC)* (2017), 1–4.

19. Aston, P., Nandi, M., Christie, M. & Huang, Y. *Comparison of Attractor Reconstruction and HRV Methods for Analysing Blood Pressure Data* in. **41** (Sept. 2014).

20. Charlton, P. *et al. Measurement of Cardiovascular State Using Attractor Reconstruction Analysis* in (Aug. 2015).

21. Sauer, T., Yorke, J. A. & Casdagli, M. Embedology. *Journal of statistical Physics* **65,** 579–616 (1991).

22. Mañé, R. in *Dynamical Systems and Turbulence, Warwick 1980* 230–242 (Springer, 1981).

23. Takens, F. in *Dynamical systems and turbulence, Warwick 1980* 366–381 (Springer, 1981).

24. Ruelle, D. Strange attractors. *Turbulence, Strange Attractors, and Chaos* **16,** 195 (1995).

25. Pecora, L. M., Moniz, L., Nichols, J. & Carroll, T. L. A unified approach to attractor reconstruction. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **17,** 013110 (2007).

26. Abarbanel, H. D., Brown, R., Sidorowich, J. J. & Tsimring, L. S. The analysis of observed chaotic data in physical systems. *Reviews of modern physics* **65,** 1331 (1993).

27. Aston, P. J., Christie, M. I., Huang, Y. H. & Nandi, M. Beyond HRV: attractor reconstruction using the entire cardiovascular waveform data for novel feature extraction. *Physiological measurement* **39,** 024001 (2018).

Appendix

The codes used in this thesis can be found on the public reopsitory:

`https://github.com/edko-b/Diploma_Thesis`

The immutability of the scripts is guaranteed by hash codes.

Table 10

Script Name	*Hash*
control_select.py	8deb3e7a3b059f4c9f4a6b415391205f
method1_complete.py	344be08188b797340e2b966465e0b2d7
method2_prepare_dataset.py	4d00fe0b92e601344ae90a34ca4fc58a
method2_analysis.py	544df7f044c259985262db22c017340a
method3_prepare_dataset.py	16629e8f2271d604a1ad881140784eeb
method3_find_optimal.py	0187cf6d0d20ce07122ae8198a096f95
method3_analysis.py	3a0020b7766eeeeea95dd21a9a36fa9b