CHALMERS

GÖTEBORGS UNIVERSITET

Maskininlärning för diagnosticering av perifer neuropati

med icke-parametriska klassificeringsmetoder

Examensarbete för kandidatexamen i matematik vid Göteborgs universitet
Kandidatarbete inom civilingenjörsutbildningen vid Chalmers

Margareta Carlerös
Nina Malmqvist
Josefin Nilsson
Fredrik Skärberg

Institutionen för Matematiska vetenskaper
CHALMERS TEKNISKA HÖGSKOLA
GÖTEBORGS UNIVERSITET
Göteborg, Sverige 2019

Maskininlärning för diagnosticering av perifer neuropati

med icke-parametriska klassificeringsmetoder

Examensarbete för kandidatexamen i matematisk statistik vid Göteborgs universitet
Fredrik Skärberg Margareta Carlerös

Kandidatarbete i matematik inom civilingenjörsprogrammet Bioteknik vid Chalmers
Josefin Nilsson

Kandidatarbete i matematik inom civilingenjörsprogrammet Teknisk matematik vid Chalmers
Nina Malmqvist

Handledare: Aila Särkkä
Anders Hildeman

Examinator: Ulla Dinger
Maria Roginskaya

Institutionen för Matematiska vetenskaper
CHALMERS TEKNISKA HÖGSKOLA
GÖTEBORGS UNIVERSITET
Göteborg, Sverige 2019

Populärvetenskaplig presentation

Nya möjligheter för AI vid diagnosticering av nervsjukdomar

Med hjälp av artificiell intelligens kan nu behövande patienter få sin diagnos för nervsjukdomar redan i tidigt skede. Detta bara genom att läkaren tar en bild på patientens hud. Nervsjukdomar är många gånger en följd av diabetes och upptäcks ofta inte förrän det är långt gånget. Risken är då stor att patienten får svåra sår på fötter eller i värsta fall behöver genomgå en fotamputation. När de väl fått sin diagnos kan dock förloppet bromsas. Utvecklingen går framåt - inte minst hos sjukhusen.

En förutsättning för att ge rätt vård är att kunna diagnosticera patienter med en viss sjukdom. Problemet med många nervsjukdomar är dock att de utvecklas relativt långsamt och patienten märker det ofta inte själv förrän sent i sjukdomsförloppet. Perifer neuropati är en av de nervsjukdomar som kan komma som följd av diabetes, men även vid cellgiftsbehandlingar av cancer. Sjukdomen uppkommer först i fötterna där nerverna sakta bryts ned. Detta gör att den sjuka personen får svårt att röra på sig och hålla balans. Eftersom känseln i fötterna kan helt försvinna finns det stor risk att patienten får allvarliga sår vilket skulle kunna leda till fotamputation. Något som också märks är att huden tappar sin normala förmåga att svettas, och det är detta som en forskargrupp vid University of Minnesota har spunnit vidare på.

Svettningarna kan mätas med hjälp av en liten kamera som fästs på foten eller vaden och helt enkelt filmar en liten sekvens av när huden utsätts för svettningar. Detta ska då ge ett nytt sätt att upptäcka symptomen än de svett- och känseltesterna som görs nu vid diagnosticering av dessa sjukdomar. Eftersom svettningarna inte går att se tydligt med ögat kan en dator användas för att se mönster och på så sätt säga om patienten är sjuk eller inte.

- En artificiell intelligens blir aldrig trött och kan jobba på obekväm arbetstid, säger Max Gordon, överläkare och forskare i ortopedi [1]. Gordon menar att framtiden inom diagnosticering mycket väl kan innefatta till största del tekniska lösningar. Dels för att få säkrare resultat utan mänskliga fel, dels för att spara tid. Förbättringar i sjukvården är ett ständigt arbetsområde då bland annat väntetiderna behöver förkortas. Vad gäller detta projekt med undersökningar av svettkörtlar och deras mönster hade det inte ens gått att göra utan en slags intelligent dator.

Datorn kan se hur värden från en patient liknar en annan patient baserat på flera hundra testpersoner. Genom att lägga in resultat från kameran i datorn kan den säga med hur stor sannolikhet patienten är sjuk. Det har testats flera olika sätt att lära datorn att känna igen mönster i bilderna. Baserat på flera värden så som hur stor del av bildytan som är täckt av svett och medelstorleken av svettfläckarna. Vi är dock fortfarande i ett tidigt stadium i denna teknik för diagnosticering och fler studier kommer behövas med insamling av data från patienter. Mer data kommer ge ett säkrare resultat och på så vis kan tekniken också bli en del av sjukvårdens nya rutiner. Förhoppningsvis kommer vi se en framtid där fler patienter kan få möjlighet att upptäcka sina nervsjukdomar i tid. Därtill också en framtid där vi kan få en effektivare sjukvård där läkare kan använda sin tid åt mer relevanta saker än det som faktiskt en artificiell intelligens kan göra bättre.

[1] *Medicinsk Vetenskap nummer 1 2017*

Sammanfattning

Den här rapporten undersöker möjligheten till att diagnosticera perifer neuropati med hjälp av icke-parametriska klassificeringsmetoder. Perifer neuropati är ett sjukdomstillstånd som kännetecknas av skador på nerverna längst ut i nervsystemet med symptom som först uppkommer i fötterna och senare i vaderna. Datan som är använd i detta projekt kommer ifrån Dr. William Kennedys forskargrupp vid University of Minnesota. Totalt innehåller datan 401 observationer från 120 friska kontroller samt 65 personer med förmodad perifer neuropati till följd av cellgiftsbehandling (varav 18 bekräftats ha perifer neuropati genom andra undersökningar). Datan är insamlad med hjälp av ett dynamisk svettest, en ny diagnostisk metod för att upptäcka onormal svettning och därmed perifer neuropati. Vi jämför i detta projekt tre olika maskininlärningsmetoder för att klassificera försökspersoner som sjuka (perifer neuropati) eller friska (inte perifer neuropati): k-NN, slumpmässig skog och neurala nätverk. Dessa metoder skiljer sig åt i dess komplexitet, alla med olika för- och nackdelar. För att utvärdera vilken klassificeringsmetod som är bäst så utfördes en korsvalidering med en modifierad variant av Cohen's kappa. Vilken klassificeringsmetod som är bäst beror på vilket mätområde datan kommer ifrån, antingen fot, vad eller fot och vad kombinerat. Bästa klassificeringsmetoden visade sig vara slumpmässig skog, detta för vadmätningarna och där kovariaterna väljs med stegvis bakåtselektion. Denna metod ger rätt klassificering av 67% av de sjuka försökspersonerna och 96% av de friska försökspersonerna. För den bästa modellen tränad på fotmätningar så klassificeras de flesta obekräftat sjuka som sjuka medan för bästa modellen tränad på vadmätningarna så klassificeras de flesta obekräftat sjuka som friska. Detta kan indikera att symptomen av perifer neuropati uppkommer först i fötterna, vilket är något som har observerats i den kliniska verkligheten.

Nyckelord; Klassificering; AI; Statistiskinlärning; k-NN; Slumpmässig skog; Neurala nätverk; Medicinsk diagnostik; Dynamiskt Svettest

Abstract

This report investigates the possibility of diagnosing peripheral neuropathy with the help of non-parametic classification methods. Peripheral neuropathy is a disease state characterized by damage on the nerves furthest out in the nervous system, with symptoms first occuring in the feet. The data used in this project comes from Dr. William Kennedys research group at University of Minnesota. The data contains 401 observations of 120 healthy controls and 65 individuals with presumed peripheral neuropathy due to chemotherapy, (where 18 individuals have been confirmed having peripheral neuropathy through other examination procedures). The data is collected with a dynamic sweat test, a new diagnostic method to discover unusual sweating patterns and therefore also peripheral neuropathy. In this project we compare three different machine learning methods to classify subjects as sick (peripheral neuropathy) and healthy (no peripheral neuropathy): k-NN, random forest and neural networks. These methods differ in their complexity, all with their disadvantages and advantages. To evauluate which classification method that works the best a cross-validation was performed, with a modified version of Cohen's kappa. How good these classification methods perform depends on which measuring area the data comes from, either foot, calf or foot and calf combined. The best classification method was shown to be random forest, this for the calf-measurements where the covarariates are chosen by backward stepwise selection. This method correctly classifies 67% of the sick individuals and 96% of the healthy controls. With the best model trained on foot-measurements most undetermined sick individuals are being classified as sick, while for the best model trained on calf-measurement most of the undetermined sick individuals are classified as healthy. This could hint towards that the symptoms of peripheral neuropathy first appears in the feet, something that is in line with the clinical reality.

Keywords; Classification; AI; Statistical learning; k-NN; Random forest; Neural networks; Medical diagnostics; Dynamic Sweat Test

Innehåll

Förord

En loggbok (dagbok) har först efter varje möte inom gruppen och efter möte med handledare. Denna dagbok ligger som en docx. fil i SVN. Individuella tidsloggar har förts online där information om de enskilda medverkandets prestationer återfinns.

Gemensamma bidrag

Vi har haft gemensamma möten veckovis under vårterminen där alla har varit närvarande vid majoriteten av mötena. Vi har också haft möten med handledarna veckovis där vi har fått feedback och återkoppling på arbetet. De flesta simuleringar har gjorts tillsammans inom gruppen. Vi har alla läst på allmänt om perifer neuropati och klassificeringsmetoder.

Individuella bidrag

De individuella bidragen beskrivs nedan och i tabell 1, där huvudansvarig författare av avsnitt i denna rapport redovisas.

Margareta

Jag har främst läst in mig på perifer neuropati, nästlad korsvalidering, klassificeringsmått samt olika paket i R. Utöver detta har jag skrivit R-koden i detta projekt samt varit ansvarig för dagboken 2019-01-31 - 2019-03-15.

Fredrik

Jag har läst på om olika klassificeringsmetoder, framförallt slumpmässig skog. Förutom de simuleringar vi har gjort i grupp har jag individuellt gjort körningar för neurala nätverk. Har också sammanställt resultatet från dessa simuleringar. Har fixat mycket administrativt med SVN och latex.

Nina

I det här projektet har jag läst på om klassificeringsmetoder, men har framför allt fokuserat på neurala nätverk. Utöver simuleringarna tillsammans i gruppen har jag simulerat resultat för obekräftat sjuka. Jag har varit ansvarig för dagboken 2019-03-19 - 2019-05-16.

Josefin

Framförallt har jag läst på om klassificeringsmetoder och maskininlärning i stort. Jag har fokuserat på k-NN och hur metoden implementeras i R.

Tabell 1: Huvudansvarig författare av avsnitt

Avsnitt	**Margareta**	**Nina**	**Josefin**	**Fredrik**
Populärvetenskaplig presentation			X	
Sammanfattning				X
Abstract				X
1 Inledning	X			
1.1 Syfte	X	X	X	X
1.2 Frågeställningar	X	X	X	X
1.3 Avgränsningar	X	X	X	X
1.4 Rapportens struktur	X			
2 Bakgrund	X			
2.1 Perifer neuropati	X			
2.2 Dynamiskt svettest	X			
2.3 Data	X			
2.4 Kovariater				X
3 Teori	X			
3.1 Klassificeringsmetoder	X			
3.1.1 K-närmaste-grannar			X	
3.1.2 Slumpmässig skog				X
3.1.3 Neurala nätverk		X		
3.2 Träning, validering och testning av modeller	X			
3.2.1 Korsvalidering	X			
3.2.2 Nästlad korsvalidering	X			
3.3 Variabelselektion	X			
3.4 Sannolikhetströskelvärden	X			
3.5 Klassificeringsmått	X			
4 Metod	X			
4.1 Implementation av klassificeringsmetoder		X		
4.1.1 k-NN		X		
4.1.2 Slumpmässig skog		X		
4.1.3 Neurala nätverk		X		
4.2 Utvärdering och jämförelse av klassificeringsmetoder	X			
4.2 Klassificering av obekräftat sjuka			X	
5 Resultat				X
5.1 Utvärdering och jämförelse av klassificeringsmetoder				X
5.2 Klassificering av obekräftat sjuka				X
5.2.1 Jämförelse av modeller för fot och vad				X
6 Diskussion	X	X	X	X
7 Slutsats				
Bilagor	1-6			7-10

Ord- och förkortningslista

Svensk översättning	Engelskt originaluttryck	Förkortning
Beslutsnod	Decision node	
Beslutsträd	Decision tree	
Datamängd	Dataset	
Dynamiskt svettest	Dynamic sweat test	DST
Förvirringsmatris	Confusion matrix	
Indelning	Fold	
K-närmaste-grannar	K-nearest neighbors	k-NN
Korsvalidering	Cross-validation	
Neurala nätverk	Neural networks	ANN
Nästlad korsvalidering	Nested cross-validation	
Orenhetsmått	Impurity measure	
Slumpmässig skog	Random forest	
Stegvis bakåtselektion	Backward stepwise selection	
Stegvis framåtselektion	Forward stepwise selection	
Testmängd	Test set	
Träningsmängd	Training set	
Valideringsmängd	Validation set	
Viktnedbrytning	Weight decay	

1 Inledning

Medicinsk diagnostik används för att identifiera sjukdomstillstånd hos patienter och spelar därmed en central roll i vården. Detta beslutsverktyg avgör både vilka patienter som betraktas som sjuka respektive friska samt vilken behandling eller ytterligare undersökning de sjuka patienterna bör få. Diagnosticering av ett sjukdomstillstånd bygger på att de friska och sjuka patienterna skiljer sig åt med avseende på en eller flera mätbara faktorer, även benämnda kovariater. Användning av kovariater för att kategorisera en observation (eller patient) kallas klassificering och förekommer inom vitt skilda tillämpningsområden såsom att upptäcka bankbedrägeri eller dela in kunder i grupper utifrån vilket reklambudskap de är känsligast för [1]. Under de senaste decennierna har en mängd olika klassificeringsmetoder utvecklats inom maskininlärning, där en dator tränas i att känna igen kategorierna.

I detta projekt undersöks olika klassificeringsmetoder för diagnosticering av perifer neuropati, ett sjukdomstillstånd som kännetecknas av skador på nerverna längst ut i nervsystemet. Symptomen börjar i fötterna och vandrar sedan upp i benen och kan även påverka händerna [2][3]. Den vanligaste orsaken till perifer neuropati är diabetes, men även cellgiftsbehandling och vissa sjukdomar såsom HIV kan ge upphov till tillståndet [3][2]. I en typ av perifer neuropati skadas de autonoma nervfibrerna, som styr icke-viljestyrda funktioner i kropppen såsom svettning. Skador på de autonoma nerverna kan därför upptäckas genom att studera svettsekretion, där minskad svettning tyder på att de autonoma nerverna skadats [4].

Dynamiskt svettest är en ny diagnostisk metod för upptäckt av onormal svettning och därigenom perifer neuropati. I testet används en speciell kamera för att skapa en videosekvens av svettsekretionen på en liten hudyta. Utifrån videosekvensen erhålls bilder vid specifika tidpunkter, där olika kovariater sedan beräknas utifrån svettmönstret i varje bild.

Datan i detta projekt har genererats med dynamiskt svettest och är insamlad av Dr. William Kennedys forskargrupp vid University of Minnesota. Friska försökspersoner samt försökspersoner med förmodad perifer neuropati till följd av cellgiftsbehandling (där en andel bekräftats ha perifer neuropati genom andra undersökningar) har testats. Mätningar har gjorts på fot, vad eller både fot och vad.

Vi kommer i detta projekt att jämföra icke-parametriska maskininlärningsmetoder för att klassificera försökspersoner som sjuka (perifer neuropati) eller friska (inte perifer neuropati). Icke-parametriska maskininlärningsmetoder antar inte att datan är fördelad enligt en viss sannolikhetsfördelning, såsom t.ex. en normalfördelning [5], vilket gör dem fördelaktiga då tidigare kunskap om datan är begränsad [6].

1.1 Syfte

Syftet med projektet är att undersöka om det är möjligt att bedöma om en patient är frisk eller lider av perifer neuropati genom att analysera deras svettmönster. För att göra detta kommer data från friska patienter och patienter som lider av perifer neuropati analyseras med hjälp av olika klassificeringsmetoder som tränas och optimeras. Projektet kommer baseras på metoderna k-närmaste-grannar (k-NN), slumpmässig skog och neurala nätverk. Klassificeringsmetoderna utvärderas sedan med avseende på deras förmågor att skilja på friska och sjuka patienter utifrån mätningar gjorda på vad, fot samt fot- och vadmätningar kombinerat.

1.2 Frågeställningar

Nedan följer frågeställningarna vi ska besvara

- Är det med hjälp av vald klassificeringsmetod möjligt att identifiera friska och sjuka patienter?
- Vilken klassificeringsmetod fungerar bäst på respektive mätområde? Hur kan denna klassificeringsmetod optimeras?

- Vilket mätområde ger bäst klassificering, d.v.s. var är det enklast att se skillnad på friska och sjuka individer?
- Vilka kovariater är mest användbara? Hur skiljer sig detta beroende på mätområde?

1.3 Avgränsningar

Perifer neuropati anses vara ett kroppssymmetriskt tillstånd. Datan som projektet utgår ifrån stödjer det här då ingen tydlig skillnad kan identifieras beroende på om mätningen utfördes på höger respektive vänster sida av kroppen (se bilaga 1). Därför tas ingen hänsyn till vilken sida mätningen utfördes på. Ingen hänsyn tas heller till vilket datum eller vilken tid på dagen som mätningen genomfördes. Tillgång till rådata saknas och därför kommer inga ytterligare kovariater räknas ut.

Det finns ett flertal klassificieringsmetoder och därmed behövs en avgränsning göras med hänsyn till vilka metoder som den här rapporten ska hantera. Rapporten kommer att beröra de tre icke-parametriska metoderna k-NN, slumpmässiga skogar och neurala nätverk. Neurala nätverk kan defineras som både en parametrisk och icke-parametrisk metod, men vanligtvis benämns neurala nätverk som en icke-parametisk metod. Vi har valt att inrikta oss på de här metoderna då de har visats vara tillförlitliga vid klassficieringsproblem, samt ett egenintresse till att analysera just de här metoderna.

1.4 Rapportens struktur

Denna rapport är strukturerad på följande sätt: i kapitel 2 beskrivs tillståndet perifer neuropati samt en ny metod för diagnosticering av perifer neuropati, dynamiskt svettest. Mätningar som gjorts med dynamiskt svettest är utgångspunkten för detta projekt. Denna data presenteras i kapitel 2.

Därefter beskrivs i kapitel 3 principen för de icke-parametriska klassificeringsmetoder som vi avser att undersöka följt av hur dessa kan utvärderas. Olika aspekter relaterade till klassificeringsmodellers konstruktion och utvärdering redogörs dessutom för.

Specifika metodval beskrivs och motiveras i kapitel 4. Vi går här igenom vilka befintliga implementationer av klassificeringsmetoderna vi använder samt hur metoderna kommer att utvärderas per mätområde (fot, vad eller både fot och vad). Hur den bästa klassificeringsmetoden per mätområde väljs beskrivs också. Till sist presenteras hur en modell skapas utifrån den bästa klassificeringsmetoden per mätområde för klassificering av försökspersoner med misstänkt (men ej bekräftad) perifer neuropati.

Resultatet består av två huvuddelar (se kapitel 5). I den första delen presenteras hur väl de olika klassificeringsmetoderna fungerar per mätområde och den bästa klassificeringsmetoden per mätområde identifieras. I den andra delen redovisas klassificeringen av försökspersoner med misstänkt (men ej bekräftad) perifer neuropati på de olika mätområdena och modellerna som skapats per mätområde jämförs.

Den tillhandahållna datan, metodval, resultat samt framtida forskningsinriktningar diskuteras till sist i kapitel 6.

2 Bakgrund

I detta kapitel ges i avsnitt 2.1 en översikt av de olika formerna av perifer neuropati vartefter sjukdomstillståndets utbredning och konsekvenser påpekas. Vikten av tidig diagnostik diskuteras sedan och behovet av nya diagnostiska metoder framhålls. Därefter beskrivs i avsnitt 2.2 dynamiskt svettest, en ny metod för att upptäcka en underkategori av perifer neuropati som kännetecknas av skador på autonoma nerver. Denna metod har testats i ett antal tidiga studier på försökspersoner.

Det är data från en av dessa studier som är utgångspunkten för detta projekt och beskrivs i avsnitt 2.3. I datan ingår ett antal mätvärden, så kallade ”kovariater”, som beräknats baserat på resultatet från varje dynamiskt svettest. Hur kovariaterna beräknats förklaras i avsnitt 2.4. Datan och dess kovariater kommer i detta projekt att användas för att konstruera modeller för upptäckt av perifer neuropati, vilket redogörs för i senare kapitel.

2.1 Perifer neuropati

Flera olika typer av nerver kan skadas vid perifer neuropati. Skador på de autonoma nerverna kan märkas bl.a. genom minskad svettning, eftersom autonoma nerver styr sekretion av svett från svettkörtlarna [7]. Utöver de autonoma nerverna kan även funktionen hos sensoriska och motoriska nerver påverkas [3]. Skador på sensoriska nerver medför nedsatt känsel medan skador på motoriska nerver innebär att t.ex. muskler förtvinar [8]. Vanligtvis är symptomen kroppssymmetriska, d.v.s. symptomen uppkommer i båda kroppshalvorna samtidigt och i lika stor utsträckning [3][2]. I regel drabbas de tunnare nervfibrerna tidigare än de tjockare nervfibrerna [8].

Perifer neuropati kan leda till svåra smärtor [9]. Hos diabetespatienter med perifer neuropati uppstår även ofta fotsår som, om de förblir obehandlade, i värsta fall kan leda till fotamputation [8]. Det uppskattas att 30-66% av diabetespatienter utvecklar perifer neuropati [2]. Globalt sett är diabetes och dess komplikationer ett stort och växande problem då den globala prevalensen av diabetes uppskattades till 425 miljoner människor 2017 och förväntas öka till 628 miljoner 2045 [9].

Även om diabetes är den vanligaste orsaken till perifer neuropati så är även cellgiftsbehandling en ökande orsak till utveckling av tillståndet. I takt med att fler cancerpatienter blir botade är det fler patienter som lever med biverkningar av cancerbehandling. Att förebygga tillståndet är svårt eftersom cancerbehandlingen då måste anpassas vilket kan påverka patientens långtidsöverlevnad [10]. Förekomsten av perifer neuropati till följd av cellgiftsbehandling har beräknats till mellan 19% och 85%, beroende på vilken läkemedelssubstans och behandlingsplan som använts [11]. Vid cellgiftsbehandling är skador på sensoriska nerver vanligare än skador på autonoma och motoriska nerver [10][12].

Allmänt anses perifer neuropati vara ett irreversibelt tillstånd även om man hos diabetespatienter observerat att förändringar på tunnare nervfibrer i högre grad är reversibla än förändringar på tjockare nervfibrer [4]. I de fall där förebyggande behandling finns är det därför viktigt att screena patienter som tillhör riskgrupper för att utveckla perifer neuropati regelbundet för att snabbt kunna sätta in förebyggande behandling och minimera risken för komplikationer [8][13].

Dagens standardundersökningar för perifer neuropati är främst inriktade på att upptäcka skador på sensoriska och motoriska nerver [4]. Detta kan utgöra ett problem för vissa patienter där de autonoma nerverna drabbas selektivt [7][4]. Därför är en metod som möjliggör studier av funktionen hos autonoma nervfibrer ett viktigt komplement till nuvarande diagnostiska tester.

2.2 Dynamiskt svettest

Undersökning av svettsekretion för upptäckt av autonom perifer neuropati utgör ett icke-invasivt och därmed för patienten skonsammare alternativ till hudbiopsi, som innebär att en bit hudvävnad tas från patienten vartefter nervernas struktur studeras i mikroskop [14]. Flera metoder för undersökning av autonoma nerver via svettsekretion har utvecklats [2][9]. Dock ger de nuvarande metoderna inte tillräckligt detaljerad information och möjliggör därför inte studier av små autonoma nervfibrer i huden [15].

Dynamiskt svettest (dynamic sweat test, DST) är en metod som löser detta problem genom att använda en speciell videokamera som ger en högupplöst film av svettsekretionen på en hudyta av storlek 2x2 cm under 1 minut [15]. Innan videoupptagningen börjar, appliceras läkemedlet pilokarpin på huden som passerar igenom huden då en lätt ström (2 mA) tillförs under 5 minuter. Ämnet stimulerar de autonoma nerverna i huden vilket leder till svettning. Från videosekvensen

selekteras bilder tagna vid specifika tidpunkter. Utifrån dessa bilder kan olika kovariater räknas ut, såsom hur många svettfläckar som syns på bilden och hur stor andel av bildytan som är täckt av svettfläckar (se avsnitt 2.4). Eftersom skador på de perifera nerverna leder till minskad svettning är förhoppningen att de framräknade kovariaterna kan användas för att bygga modeller som kan upptäcka minskad svettning och därigenom perifer neuropati.

2.3 Data

Datan i detta projekt har genererats med hjälp av dynamiskt svettest och har samlats in av Dr. William Kennedys forskargrupp vid University of Minnesota mellan den 5 september 2012 och den 22 december 2014. Totalt ingår 401 observationer från 120 friska kontrollpersoner som inte lider av perifer neuropati och 65 personer med förmodad perifer neuropati till följd av cellgiftsbehandling (varav 18 bekräftats ha perifer neuropati genom andra undersökningar). Inga värden saknas i datan. Per försöksperson har mätningar gjorts på fot, vad eller både fot och vad. En majoritet av mätningarna är gjorda på vad (216 av 401) och relativt få mätningar har gjorts på försökspersoner med bekräftad perifer neuropati (27 av 401, se bilaga 2 för mer information om antal mätningar per mätområde och per grupp av försökspersoner). På en majoritet av försökspersonerna (135 av 185) har mer än en mätning gjorts (se bilaga 2).

I den tillhandahållna datan ingår information om 23 olika variabler. Av dessa variabler är 15 kovariater, som presenteras i avsnitt 2.4. En kort beskrivning av variablerna förutom kovariaterna som ingår i datan återfinns i bilaga 3.

I efterföljande text kommer friska kontroller att benämnas "friska", försökspersoner med bekräftad neuropati som "bekräftat sjuka" och försökspersoner med misstänkt men obekräftad neuropati som "obekräftat sjuka". Vidare kommer hänvisningar till "datamängden" att gälla allmänt för datamängden av fotmätningar och datamängden av vadmätningar, om ej närmare specificerat. Då alla observationer (oberoende av mätområde) avses, kommer dessa att benämnas "hela datamängden".

2.4 Kovariater

Datan i detta projekt består av bilder tagna vid tidpunkterna 1, 10 och 30 sekunder med hjälp av ett dynamiskt svettest. Från dessa bilder har svettmönstret analyserats och fem stycken olika mått räknats ut. Då bilderna är tagna vid tre olika tidpunkter betyder det att vi totalt har 15 stycken mått per mätning, dessa 15 mått kallar vi för kovariater. De mått som har räknats ut per bild är följande:

- **WAF_f*:** Andel av bildytan täckt av svett.
- **Intensity_f*:** Antalet svettfläckar i en bild.
- **Avesize_f*:** Medelstorleken av svettfläckarna mätt i pixlar.
- **CI300_f*:** Ett mått på grupperingar i datan, har beräknas genom att integrera den skattade L-funktionen från bilderna.
- **Hazard_mode_f*:** Ett estimerat mått på tomrummet mellan svettfläckar.

Här står * för tidpunkterna 1, 10 och 30 sekunder. I varje bild har man lokaliserat centroiden av alla svettfläckar och noterat dem med en unik koordinat (x, y). Därefter har arean av denna svettfläck beräknats. Måttet **Avesize_f*** beräknas således genom att ta medelvärdet av alla svettfläckars area. **WAF_f*** beräknas genom att summera alla svettfläckars area och dela med storleken på bildytan. Antalet svettfläckar, **Intensity_f***, beräknas genom att räkna hur många centroider som lokaliserats i bilden. I det fall då två svettfläckar har gått ihop, vilket de kan göra mellan två tidpunkter, definieras en ny punkt (x_2, y_2) för dess centroid. Den nya svettpunktens area definieras då som summan av de två svettfläckarnas area.

Ett mått på grupperingar i datan, **CI300_f***, fås genom att beräkna följande integral

$$LCI(R) = \int_0^R L_c(r)dr \tag{1}$$

där R antingen är 300 eller 600 pixlar. $L_c(R)$ är en centrerad version av Ripley's K-funktion som också är variansstabiliserad. Ripley's K-funktion definieras som

$$K(r) = \lambda^{-1}\mathbf{E}\big[\text{antal svettfläckar inom distansen } r \text{ till given svettfläck}\big] \tag{2}$$

och $L_c(R)$ som

$$L_c(r) = \sqrt{\frac{K(r)}{\pi}} - r \tag{3}$$

Då $L_c(R)$ är lika med 0 vi har full spatial slumpmässighet (Poisson process). Värden mindre än 0 indikerar regularitet medan värden större än 0 indikerar existensen av grupperingar i datan.

Hazard_mode_f* är beräknat genom att först ta reda på den fördelningsfunktion som beskriver distansen från en godtycklig punkt (x, y) till närmaste svettfläck. Låt D beteckna denna distans, vi har då fördelningsfunktionen

$$F(r) = \mathbf{P}(D \leq r) \tag{4}$$

med motsvarande hasardfunktion

$$h(r) = \frac{f(r)}{1 - F(r)} \tag{5}$$

där f är täthetsfunktionen motsvarande F. Vi kommer att titta på den distans r som maximerar hasardfunktionen, d.v.s. det mest sannolika avståndet till närmaste svettfläck. Detta kommer att vara vårat mått, **Hazard_mode_f***.

3 Teori

Vi jämför i detta projekt icke-parametriska maskininlärningsmetoder för klassificering av friska respektive bekräftat sjuka försökspersoner. Klassificering med hjälp av maskininlärning innebär att en mängd data med känd klassificering (d.v.s. vi vet redan om en försöksperson är frisk eller sjuk) först görs tillgänglig för en dator. Därefter lär sig datorn att känna igen de olika kategorierna genom att tillämpa en viss princip och förinställda parametrar. Här är principen det som brukar kallas klassificeringsmetod. Själva lärandeprocessen, där modellen anpassas till den tillhandahållna datamängden, beskrivs ofta som att en klassificeringsmodell "tränas". Det är den tränade modellen som sedan används för att klassificera nya observationer (d.v.s. nya försökspersoner eller patienter).

De klassificeringsmetoder som vi avser att undersöka i detta projekt beskrivs i avsnitt 3.1. Per klassificeringsmetod redogörs för den princip som används för att skapa en modell samt vilka parametrar som ställs in innan modellen tränas.

En central aspekt i valet av parametrar är den resulterande modellens komplexitet. Målet är att konstruera en modell som bäst klassificerar nya osedda observationer. En modell som är för komplex kommer att överanpassas till datan som används för träning vilket ger stor osäkerhet (hög varians) då en ny observation klassificeras. Emellertid leder en alltför förenklad modell till en undermålig approximation av verkligheten (hög bias). Vi vill alltså att modellen ska vara tillräckligt komplex för att fånga upp signalen i datan men tillräckligt förenklad för att undvika att påverkas av bruset i datan. Givet en viss klassificeringsmetod är det dock inte möjligt att minimera varians och bias samtidigt. För att minimera felet som modellen gör måste en avvägning göras mellan varians och bias. Denna balansgång brukar benämnas "bias-variance tradeoff" [16].

Eftersom vi t.ex. inte vet hur verkligheten ser ut, är en exakt bestämning av bias (och därmed

felet), svårt. En uppfattning om hur väl modellen klassificerar nya observationer kan erhållas genom att först slumpmässigt dela in observationer med känd klassificering i två grupper, en "träningsmängd" (training set) och en "valideringsmängd" (validation set) [16]. Träningsmängden används för att träna en modell som sedan klassificerar observationerna i valideringsmängden. Genom att jämföra modellens klassificering med den verkliga klassificeringen av valideringsmängden, kan ett mått på hur väl modellen klassificerar ny data räknas ut. Detta mått kan därefter användas som utgångspunkt för att välja bland modeller.

När en modell selekterats kan den utvärderas genom att testas på nya observationer med känd klassificering som kallas "testmängd" (test set). Det kan vara en utmaning att dela in datan i tränings-, validerings- och testmängd på ett representativt sätt för små datamängder [16][17]. Eftersom datan i detta projekt består av relativt få observationer kommer därför en metod som heter "nästlad korsvalidering" att implementeras för att möjliggöra selektion och utvärdering av modellerna (se avsnitt 3.2).

En ytterligare faktor som avgör komplexiteten hos en modell är antal kovariater som ingår. Fler kovariater i en modell gör den mer komplex och därmed ökar risken att modellen överanpassas till datan som används för att träna den. Olika strategier för att minska antalet kovariater har utvecklats och de vi avser att använda i detta projekt beskrivs i avsnitt 3.3. Vi kommer dessutom att undersöka effekten av att ändra sannolikheten som modellen anger för en observation klassificeras som "sjuk" som förklaras i avsnitt 3.4. Slutligen beskrivs i detta kapitel klassificeringsmåttet Cohen's kappa (se avsnitt 3.5).

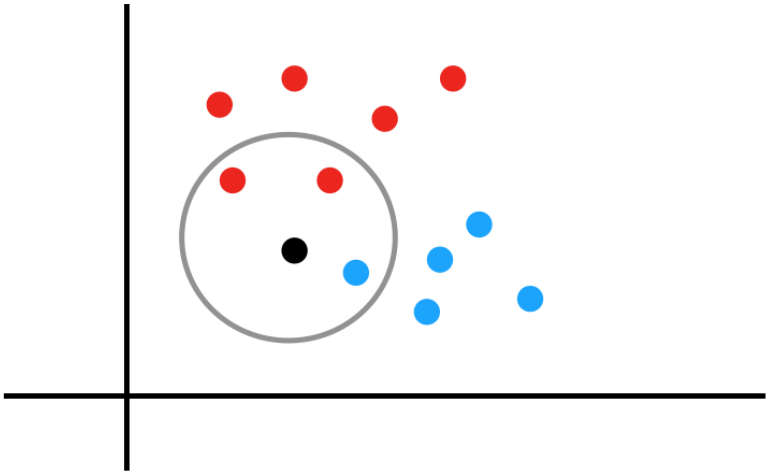
3.1 Klassificieringsmetoder

De maskininlärningsmetoder för klassificering som vi studerar i detta projekt kan alla betraktas som icke-parametriska eftersom de inte gör några antaganden om sannolikhetsfördelningen av datan. Dock utgår dessa klassificeringsmetoder från olika principer för hur klassificeringen görs och deras lämplighet beror på egenskaper hos datan. Det är däremot svårt att på förhand identifiera den lämpligaste metoden [6]. Därför är det viktigt att undersöka olika metoder för klassificering av datan.

Klassificeringsmetoderna som vi implementerar i detta projekt är k-närmaste grannar, slumpmässig skog och neurala nätverk. I k-närmaste grannar består modellen enbart av observationer med känd klassificering, där klassificeringen av en ny observation bestäms av kategorin som majoriteten av de k närmaste grannarna tillhör (se avsnitt 3.1.1). I motsats till k-NN, består slumpmässig skog av en uppsättning regler som konstrueras utifrån observationer med känd klassificering. Dessa regler används sedan för att klassificera nya observationer. Slumpmässig skog är ett exempel på en så kallad sammansatt inlärningsmetod (ensemble method) som är baserad på ett antal unika beslutsträd. Beslutsträd och slumpmässig skog beskrivs i avsnitt 3.1.2. Ännu mer komplexa regler för klassificering skapas med neurala nätverk, en metod som har ökat i popularitet de senaste åren (se avsnitt 3.1.3).

3.1.1 K-närmaste-grannar (K-nearest neighbors)

K-närmaste-grannar (k-NN) är en klassificeringsmetod som bygger på att hitta närmaste avstånden mellan olika datapunkter och på så vis klassificera dem [18]. Algoritmen tränas först på en färdigklassificerad datamängd där varje observation representeras som en punkt i ett flerdimensionellt rum [19]. För att sedan klassificera en okänd observation används ett antal, k, observationer som ligger närmast i avstånd till en ny observation [19]. I figur 2 illustreras ett enkelt exempel av algoritmen där vi har två klasser: blå och röd. Om vi vill klassificera den svarta punkten till en av färgerna och valet av k är tre, kommer de tre närmaste punkterna att ringas in. Eftersom majoriteten av observationerna i ringen är röda kommer den svarta punkten att klassificeras som röd.



Figur 1: Visualisering av k-NN där den svarta punkten klassificeras som röd vid $k = 3$

Valet av parametern k är en väldigt viktig del i denna klassificeringsmetod, men kan ofta vara svår att optimera. Vid val av k kan det vara bra att förstå vad ett högt respektive lågt värde kan få för konsekvenser. Ett lågt k begränsar regionen för en viss klassificering vilket gör att algoritmen har svårt att se hela fördelningen. Lågt k ger alltså hög flexibilitet och lågt bias men en hög varians [20]. Ett högt k kommer istället medföra fler väljare som gör att fler observationer inkluderas. Höga k har alltså lägre varians men ett större bias [20]. Vid beräkning av distansen mellan observationerna används för det mesta euklidiskt avstånd[21]. Detta avstånd beräknas enligt följande:

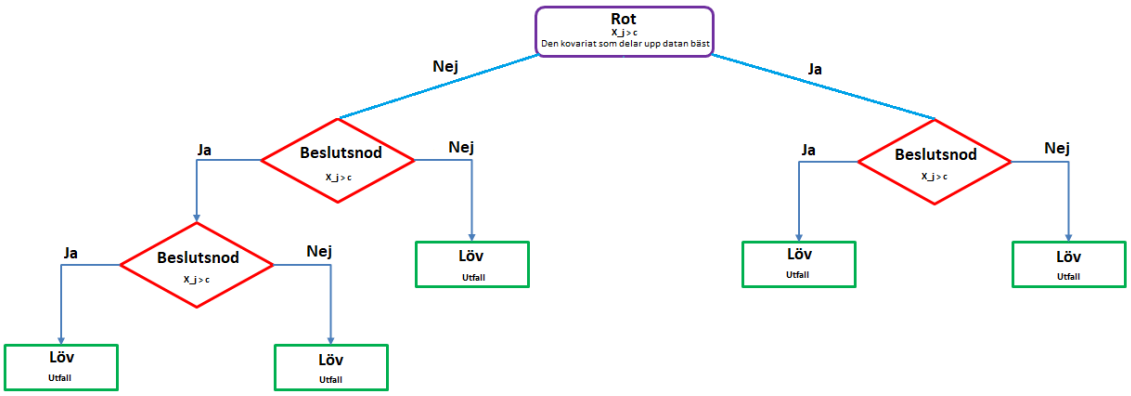
$$d(X, Y) = \sqrt{\sum_{i=1}^{n}(X_i - Y_i)^2} \tag{6}$$

$d(X, Y)$ är det euklidiska avståndet mellan punkterna X och Y och n är antalet dimensioner [21].

3.1.2 Slumpmässig skog (Random forest)

Slumpmässig skog är en klassificeringsmetod som använder sig av en mängd unika beslutsträd till att prediktera. Beslutsträd i sig utgår från en hierarkisk data-struktur, är icke-parametrisk och delas vanligtvis upp i två olika kategorier, regressions- och klassifikationsträd. Skillnaden mellan dessa är att regressionsträdets beroende variabel är kontinuerlig medan klassifikationsträdets beroende variabel är kategorisk [22].

Beslutsträdet är hierarkiskt uppbyggt av tre komponenter; rot, beslutsnoder och löv, (se figur 2). Givet en mängd kovariater byggs trädet upp genom att först välja den kovariat som delar upp datan bäst, denna kovariat kommer att vara roten i trädet. Vid nästa beslutnod evalueras en testfunktion som antar ett binärt svar, d.v.s. JA eller NEJ. Detta görs för alla kovariater och man väljer återigen den kovariat som delar upp datan bäst. Testfunktionen kan exempelvis vara $f(X_j) > c$, där c är en konstant och $f(X_j)$ är värdet på en kovariat. Detta fortgår rekursivt, och processen avslutas då en viss toleransnivå är uppfylld. För klassifikationsträd definieras en delning som ren, om vid en delning endast en klass är närvarande i lövet. Då detta sker avslutas också processen [23].



Figur 2: Schematisk bild över ett godtyckligt beslutsträd. Här är utfall den beroende variabeln, vilket för ett klassifikationträd är en klass. Vid varje beslutsnod beräknas ett mått på orenhet och en delning sker därefter.

För varje kovariat vid en beslutsnod beräknas ett mått på orenhet. Detta mått avgör vilken kovariat som delar upp datan bäst. Ju lägre mått på orenhet desto bättre är delningen. Anta att vi har ett problem med två klasser, där p är sannolikheten för klass 1 och $1 - p = q$ är sannolikheten för klass 2, efter en godtycklig beslutsnod. Det mått vi kommer att använda oss av är gini index, vilket delfinernas som

$$\phi(p, q) = 2pq \tag{7}$$

Minimum sker då endast en klass är närvarande efter delningen, då antar funktionen värdet 0. Maximum är således $2 \cdot 0.5 \cdot 0.5 = 0.5$, vilket sker då prevalensen av båda klasser efter delning är lika. Ett annat vanligt mått på orenhet är entropi, vilket definieras som $\phi(p, q) = -p \log_2(p) - q \log_2(q)$. Måttet på orenhet tar inte hänsyn till vad som har hänt innan nuvarande beslutsnod och vad som kommer att hända i beslutsträdet efter delningen. Således är endast denna delning lokalt optimal, och det finns risk till att det beslutsträd som byggs upp inte är det mest optimala.[22]

Ett problem med att endast använda ett enda beslutsträd för att klassificera är att man då är väldigt känslig för överanpassning [24]. Beslutsträdet blir bra på att klassificera den datan den tränats på men misslyckas att generalisera till ny data. Ett sätt att komma runt detta problem är att istället generera en mängd olika beslutsträd, liknande de i figur 2. Denna mängd av olika beslutsträd är vad som kallas slumpmässig skog. Dessa beslutsträd kan därefter tillsammans användas för att klassificera. Det absolut vanligaste är att man låter alla genererade beslutsträd klassificera utfallet och därefter använda sig av en majoritetsröstning.

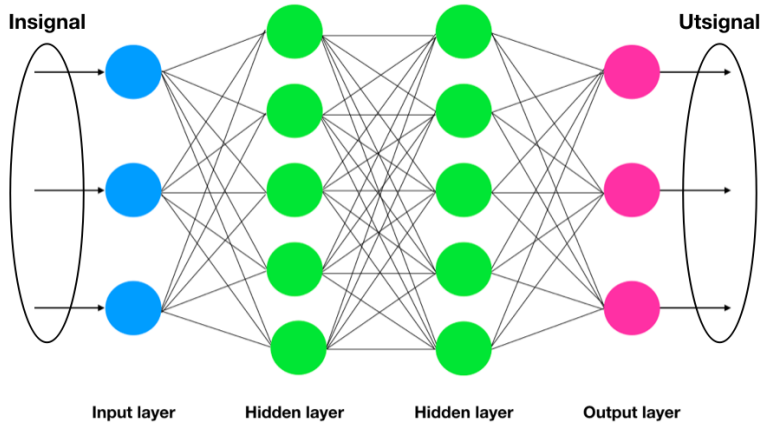
Att gå igenom och generera alla möjliga beslutsträd är oftast en alltför krävande process. Vad metoden slumpmässig skog istället gör är att slumpmässigt välja ut en delmängd av kovariater i varje steg i uppbyggandet av beslutsträdet. Det betyder att istället för att gå igenom alla kovariater vid en beslutsnod gås endast ett fåtal igenom. Detta generar en stor variation av beslutsträd, vilka tillsammans har en starkare prediktionskraft och är mer robusta. En nackdel med metoden är att den kan vara väldigt tidskrävande, vilket således beror på valet av hur många beslutsträd som ska byggas upp samt storleken av delmängden av kovariater [25].

3.1.3 Neurala nätverk (Neural networks)

Artificiella neurala nätverk (ANN) består av en mängd algoritmer där strukturen är utformad på ett sådant sätt att den efterliknar hjärnans struktur [26]. ANN är optimerad för att hitta komplexa mönster för att kunna klassificera data [27].

Strukturen för ANN består av ett inmatningslager (input layer) där en eller flera insignaler tas

emot, ett eller flera dolda lager (hidden layer) där insignalerna transformeras och ett utmatningslager (output layer) som representerar de olika utsignalerna [26]. Figur (3) visar de olika lagren i ANN.



Figur 3: Schematisk bild över ett neuralt nätverk. Insignalen passerar först genom inmatningslagret (input layer) för att sedan fortsätta genom de dolda lagren (hidden layers). Till sist klassificeras signalen i utmatningslagret (output layer).

Mängden av insignaler betecknas med S. Varje lager består av en rad noder kallade neuroner. Till en nod kommer en eller flera insignaler, x_i där varje signal viktas med en koefficient, ω_i, $i \in S$. Idén med neurala nätverk är att neuronerna sitter ihoplänkade och att vikterna ändras under inlärningen. Noderna har en aktiveringsfunktion som bestämmer i vilken grad signalen ska fortsätta i nätverket, se beskrivning av olika aktiveringsfunktioner nedan. Alla de viktade insignaler som kommer till noden summeras och denna summan passerar sedan genom nodens aktiveringsfunktion [26].

Neurala nätverk överanpassar ofta datan (overfitting) genom att den har för många vikter. En metod för att minska antalet vikter är viktnedbrytning (weight decay), vilket innebär att vikterna kommer att bestraffas så att de går mot noll. Ofta används korsvalidering för att bestämma vilka vikter som ska bestraffas [16].

En av de första varianterna av en nod är perceptron, där aktivitetsfunktionen är en stegfunktion som antar värdet noll eller ett beroende på om summan av de viktade insignalerna är större eller mindre än ett tröskelvärde, b [28]. Sambandet beskrivs i ekvation (8).

$$\text{utsignalen} = \begin{cases} 0, & \text{om } \sum_i \omega_i x_i \leq b \\ 1, & \text{om } \sum_i \omega_i x_i > b, \quad i \in S \end{cases} \tag{8}$$

Perceptronen har vidareutvecklats och en av de vanligare typerna av noder är Sigmoid neuroner [28]. Den har en kontinuerlig aktiveringsfunktion sigmoidfunktionen, även kallad logitfunktion, som antar värden mellan 0 och 1 och definieras i ekvation (9) och (10) [28] [29].

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{9}$$

$$z = \sum_i \omega_i x_i + b, i \in S \tag{10}$$

3.2 Träning, validering och testning av modeller

Ett av målen med detta projekt är att identifiera klassificeringsmetoden som bäst klassificerar försökspersonerna i datamängden som sjuka respektive friska. Detta kan uppnås med nästlad korsvalidering, som bygger på en teknik kallad "korsvalidering" (se avsnitt 3.2.1). Korsvalidering kan

användas för modellselektion eller för utvärdering av en modell, men däremot inte för båda ändamålen samtidigt. Av detta skäl används nästlad korsvalidering som består av en inre korsvalidering för modellselektion och en yttre korsvalidering för modellutvärdering (se avsnitt 3.2.2)

3.2.1 Korsvalidering

Det enklaste sättet att validera eller testa en modell (för modellselektion respektive modellutvärdering) är att dela in den givna datamängden i två delar, där den ena delen används för att träna upp modellen och den andra delen används för att validera eller testa modellen. Som påpekades i inledningen till detta kapitel kan det för små datamängder vara svårt att dela upp datamängden på ett representativt sätt. Genom att endast validera eller testa modellen på en del av datamängden erhålls dessutom en begränsad uppfattning om hur väl modellen klassificerar ny (osedd) data. För små datamängder används därför korsvalidering för att validera eller testa modeller. Användning av korsvalidering för utvärdering av modeller beskrivs härnäst (förfarandet vid validering av modeller är analogt).

Korsvalidering [30] går ut på att datamängden slumpmässigt delas upp i m stycken ungefär lika stora indelningar (folds). En av de m indelningarna sätts åt sidan vartefter modellen tränas på de återstående $m-1$ indelningarna. Därefter testas den tränade modellen på den osedda indelningen. Sedan får en annan av indelningarna anta rollen som osedd data medan övriga $m-1$ indelningar används för att träna modellen. Denna procedur upprepas tills alla indelningar använts som osedd data en gång. På så sätt fås mer information om hur modellen klassificerar osedda observationer än om endast en indelning av datamängden hade gjorts. Ett mått på hur väl klassificeringen fungerade beräknas sedan baserat på klassificeringen av alla indelningar med osedd data.

3.2.2 Nästlad korsvalidering

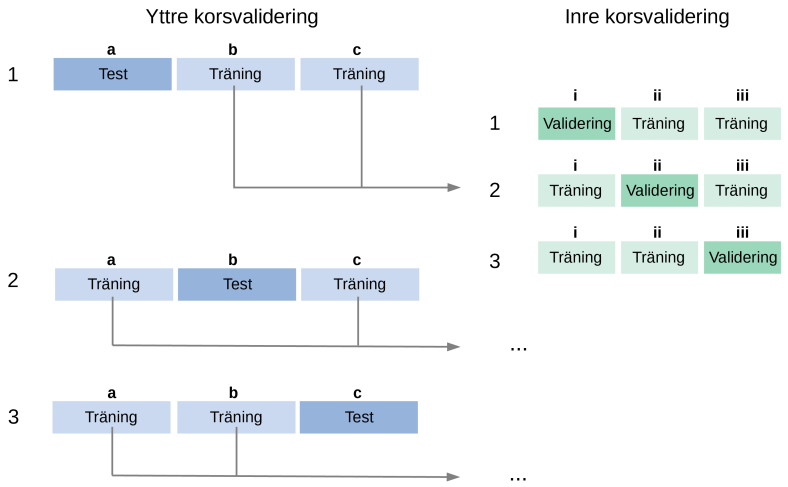
Att använda korsvalidering för att samtidigt selektera och utvärdera modeller är inte rekommenderat, eftersom utvärderingen av modellerna då tenderar att bli alltför optimistisk [31]. En variant av korsvalidering kallad nästlad korsvalidering (nested cross-validation) tillämpas ofta i dessa fall [31][17], se figur 4. I denna metod delas datamängden först upp i m stycken indelningar, där $m-1$ indelningar bildar en yttre träningsmängd och den resterande indelningen bildar en testmängd. Sammanlagt skapas m par av yttre träningsmängd med tillhörande testmängd. Varje yttre träningsmängd delas sedan i sin tur upp i n nya indelningar som bildar n stycken par av inre träningsmängder med tillhörande valideringsmängder. Ofta väljs m och n så att $m = n$ [31].

De n paren av inre tränings- och valideringsmängder utgör en så kallad inre korsvalidering som används för att selektera en modell. Varje modell som selekteras per inre korsvalidering tränas sedan om på den motsvarande yttre träningsmängden för att slutligen testas på motsvarande testmängd. De m yttre träningsmängderna och testmängderna bildar en så kallad yttre korsvalidering som används för att utvärdera modellerna.

Det är värt att notera att de m modellerna som selekteras inte nödvändigtvis är samma modell. Till exempel skulle olika inställningar på modellparametrar kunna visa sig vara optimala i de m inre korsvalideringarna. Utvärderingen av dessa m modeller på motsvarande testmängd i den yttre korsvalideringen resulterar i en väntevärdesriktig skattning av hur väl en modell (baserad på en viss klassificeringsmetod) som valts genom korsvalidering kommer att klassificera en datamängd [32]. Med nästlad korsvalidering är det därför möjligt att bestämma vilken klassificeringsmetod (av de undersökta) som är lämpligast att använda givet en datamängd.

3.3 Variabelselektion

Nära förknippat med selektion och utvärdering av modeller är valet av vilka kovariater som ska ingå i modellen, så kallad variabelselektion. Detta är viktigt att göra om datan innehåller många kovariater eftersom en modell med många kovariater riskerar att överanpassas till datamängden den tränas på. En modell med ett litet antal kovariater som alla har en stark signal är oftast att föredra framför en större modell, om de båda innehåller lika mycket av den information som är



Figur 4: Nästlad korsvalidering med tre indelningar (a, b, c) i den yttre korsvalideringen och tre indelningar (i, ii, iii) i den inre korsvalideringen. I omgång 1 av den yttre korsvalideringen antar indelning a rollen som testmängd medan resterande indelningar (b och c) bildar en träningsmängd. En inre korsvalidering utförs på träningsmängden där modeller tränas på indelningarna ii och iii och valideras med indelning i i omgång 1. Detta upprepas tills varje indelning haft rollen som valideringsmängd en gång. Baserat på de 3 valideringarna väljs en modell. Den valda modellen testas på testmängden a i omgång 1 av den yttre korsvalideringen. Detta upprepas för omgång 2 och 3 i den yttre korsvalideringen med indelning b respektive c som testmängd. Utifrån de tre valda modellernas klassificeringar av motsvarande testmängd räknas ett mått på hur väl klasificeringsmetoden klassificerar den givna datamängden ut.

relevant för klassificeringen.

Flera olika strategier har utvecklats för att bestämma vilka kovariater som har starkast signal samt hur många av kovariaterna som bör inkluderas i modellen för att ge optimal komplexitet. Ett sätt är att använda korsvalidering för att utvärdera vilka kovariater som, tillsammans med en specifik modell, ger bäst klassificering av indelningarna i korsvalideringen.

Om det totala antalet kovariater i datamängden är p så finns det $2^p - 1$ möjliga delmängder av kovariater att testa i korsvalideringen. Detta leder snabbt till många jämförelser vilket beräkningsmässigt kan bli kostsamt om p är stort. Fördelen är att delmängden kovariater som väljs är den globalt optimala (baserat på korvalideringen och den tillgängliga datamängden) för en given modell. Genom att nöja sig med att endast hitta en lokalt optimal delmängd av kovariater kan man ofta drastiskt minska beräkningstiden. Två sådana strategier är stegvis framåtselektion (stepwise forward selection) och stegvis bakåtselektion (stepwise backward selection).

Stegvis framåtselektion [30] innebär att p modeller (alla med samma värden på parametrarna) byggs baserat på var och en av de p kovariaterna. Kovariaten som ger bäst klassificering i korsvalideringen selekteras. Sedan utökas denna modell med en av de $p-1$ återstående kovariaterna. Den bästa kovariaten att utöka modellen med identifieras. Sedan jämförs denna modells klassificering i korsvalideringen med den bästa modellen från föregående steg, den reducerade modellen. Om den reducerade modellen ger bättre klassificering väljs den reducerade modellen som slutgiltig modell, annars utökas modellen på nytt med en kovariat i taget tills den reducerade modellen från steget innan uppvisar bättre klassificering. Denna reducerande modell väljs då som slutgiltig modell.

I stegvis bakåtselektion [30] börjar man tvärtom med en modell innehållande alla p kovariater och beräknar ett mått på hur väl klassificeringen fungerade utifrån korsvalideringen. Sedan byggs p modeller där en av de p kovariaterna elimineras. Bland dessa p reducerade modeller av storlek $p-1$ identifieras vilken modell som ger bäst klassificering i korsvalideringen. Om den reducerade modellen uppvisar sämre klassificering än den utökade modellen i steget innan, väljs den utökade modellen som den slutgiltiga modellen. Annars elimineras ytterligare en kovariat från den

reducerade modellen, vilket fortsätter tills den utökade modellen från steget innan ger en bättre klassificering. Denna utökade modell väljs då som slutgiltig modell.

Genom att välja kovariaterna med stegvis framåt- eller bakåtselektion och korsvalidering blir valet av kovariater anpassat till en specifik modell. Om en valbar parameter i modellen ändras är det möjligt att en annan mängd kovariater är optimal. Därför rekommenderas det att utvärdera kombinationen av kovariater och valbara parametrar tillsammans i korsvalideringen istället för att välja kovariater och parametrar separat [31].

3.4 Sannolikhetströskelvärden

Med många klassificeringsmetoder är det möjligt att skapa modeller som för varje ny observation rapporterar sannolikheten att observationen tillhör de olika klasserna, sjuk eller frisk. I regel är sannolikhetströskelvärdet 0,5, där om P(sjuk) $\geq$ 0,5 klassificeras observationen som sjuk medan om P(frisk) $<$ 0,5 klassificeras observationen som frisk.

Baserat på tillämpningen som klassificeringsmodellen är utvecklad för kan det vara aktuellt att använda ett annat sannolikhetströskelvärde än 0,5. Högre sannolikhetströskelvärden gör det svårare att klassificera en observation som sjuk, medan lägre sannolikhetströskelvärden gör det enklare att klassificera en observation som sjuk. Ett lågt sannolikhetströskelvärde skulle i en medicinsk tillämpning vara av värde då man vill screena patienter för en sjukdom och inte riskera att missa något fall. Ett högt sannolikhetströskelvärde är relevant då överrapportering av en sjukdom kan ge allvarliga konsekvenser för patienten, t.ex. vill man innan cancerbehandling påbörjas vara säker på att patienten har cancer [33].

3.5 Klassificeringsmått

Ett klasificeringsmått anger hur väl en modells klassificering överrensstämmer med den verkliga klassificeringen av observationerna i valideringss- eller testmängden. Ofta kan ett sådant mått förklaras med en förvirringsmatris (confusion matrix), se figur 5, där modellen motsvarar den nya metoden och den verkliga klassificeringen motsvarar referensmetoden [33].

		Referensmetod	
		Sjuk	Frisk
Ny metod	Sjuk	Sant positiv (sp)	Falskt positiv (fp)
	Frisk	Falskt negativ (fn)	Sant negativ (sn)

Figur 5: Förvirringsmatris för jämförelse av klassificeringar (sjuk eller frisk) erhållna med en ny metod och en referensmetod (eller verklig klassificering) där sp = antal sant positiva, fp = antal falskt positiva, fn = antal falskt negativa och sn = antal sant negativa.

Ett vanligt mått på hur väl en modells klassificering överensstämmer med den verkliga klassificeringen av observationer är tillförlitlighet (accuracy) [34] som definieras

$$\text{tillförlitlighet} = \frac{\text{sp} + \text{sn}}{\text{sp}+\text{sn}+\text{fp}+\text{fn}}, \tag{11}$$

med sp, sn, fp och fn definierade enligt figur 5.

Nackdelen med ett sådant mått är att det blir missvisande om t.ex. andelen sjuka i datamängden är låg. Med en låg andel sjuka skulle modellen kunna klassificera alla observationer som friska och fortfarande uppnå en god klassificering.

Ett mått som kompenserar för att grupperna i datamängden är olika stora är Cohen's kappa

som definieras

$$\kappa = \frac{p_{obs} - p_{slump}}{1 - p_{slump}}, \tag{12}$$

där p_{obs} är tillförlitlighet och p_{slump} är sannolikheten för slumpmässig överensstämmelse mellan modellen och den verkliga klassificeringen,

$$\begin{aligned} p_{slump} &= P(\text{sjuk}|\text{modell}) \cdot P(\text{sjuk}|\text{verklighet}) + P(\text{frisk}|\text{modell}) \cdot P(\text{frisk}|\text{verklighet}) \\ &= \left(\frac{\text{sp} + \text{fp}}{\text{sp+sn+fp+fn}}\right) \cdot \left(\frac{\text{sp} + \text{fn}}{\text{sp+sn+fp+fn}}\right) + \left(\frac{\text{fn} + \text{sn}}{\text{sp+sn+fp+fn}}\right) \cdot \left(\frac{\text{fp} + \text{sn}}{\text{sp+sn+fp+fn}}\right) \end{aligned}$$

Detta mått anger hur mycket bättre klassificeringsmodellen är än en slumpmässig klassificering [34]. En perfekt klassificering ger kappa 1, medan en slumpmässig klassificering ger kappa 0. Det är möjligt att erhålla ett negativt värde på kappa, vilket indikerar att klassificeringen är sämre än slumpmässig klassificering.

4 Metod

I detta projekt används programmeringsspråket R [35]. Vi använder i stor utsträckning befintliga funktioner i R som finns samlade i så kallade paket. Mycket av koden utgår från R-paketet `caret` där många klassificeringsmetoder finns implementerade samt många funktioner och tillval för träning och validering av modeller [36]. R-koden för detta projekt återfinns i bilaga 4.

De färdiga implementationer av klassificeringsmetoderna från `caret` som vi använder oss av specificeras i avsnitt 4.1. I detta avsnitt beskrivs även dessa implementationer kort samt vilka valbara parametrar som vi testar per klassificeringsmetod. Klassificeringsmetoderna utvärderas sedan genom nästlad korsvalidering och bästa klassificeringsmetod per mätområde identifieras (se avsnitt 4.2). Tre olika mätområden studeras: fot, vad och både fot och vad. Utifrån bästa klassificeringsmetod tränas sedan en modell per mätområde upp med alla mätningar från friska och bekräftat sjuka försökspersoner på det aktuella mätområdet. Därefter används dessa modeller för att klassificera obekräftat sjuka på motsvarande mätområde (se avsnitt 4.3). Modellerna tränade på fot- respektive vadmätningar jämförs även genom att studera modellernas klassificering av obekräftat sjuka försökspersoner som genomgått mätning på både fot och vad.

4.1 Implementation av klassificeringsmetoder

I `caret` finns ofta flera olika implementationer av samma klassificeringsmetod. Dessa skiljer sig åt i vilka ändringsbara respektive förinställda parametrar de har. Vi har valt att använda en specifik implementation per klassificeringsmetod. Implementationerna vi använder (tillsammans med de förinställda parametrarna, de valbara parametrarna samt vilka inställningar vi väljer att testa för dessa) beskrivs i avsnitt 4.1.1-4.1.3.

4.1.1 k-NN

Funktionen som används för klassificeringsmetoden k-NN är `knn` och finns i paketet `caret`. För den här versionen av k-NN är distansen inställt på euklidiskt distans och den parameter som är valbar är k.

För körningar med data från fot skapas k parametern som en vektor med udda heltal mellan 1 och 7, för vad antar k udda heltal mellan 1 och 13 och k antar udda heltal mellan 1 och 19 för fot och vad. Gränsen valdes så att inte alltid det högsta k-värdet ska väljas i den inre korsvalideringen, eftersom det då är troligt att det finns ett högre k som ger bättre resultat. Det är även betydelsefullt att inte välja ett för högt k-värde eftersom simuleringenstiden blir betydligt högre då.

4.1.2 Slumpmässig skog

Den funktion som används för slumpmässig skog är `rf` som finns i paketet `caret`. I den här versionen av slumpmässig skog finns en valbar parameter, *mtry*, som anger antal kovariater som slumpas fram i varje nod. Eftersom vi använder stegvis framåt och bakåtselektion i detta projekt varierar antalet tillgängliga kovariater i modellerna. För att undvika ett ogiltigt värde på *mtry*, vilket innebär att antal kovariater som slumpas fram är högre än antalet tillgängliga kovariater, bestäms högsta värde på *mtry* genom att multiplicera antalet tillgängliga kovariater med 40% och sedan avrunda talet uppåt. Alla heltal från 1 till detta maxvärde används som inställning för *mtry*.

I `rf` är det förinställda värdet på antal beslutsträd 500 och modellen använder Gini index som orenhetsmått. Beslutsträden vägs lika i klassificeringen av en observation.

4.1.3 Neurala nätverk

Koden använder funktion `nnet` vilket även den här kommer från paketet `caret`. Den här funktionen använder sig av feedforward och har ett dolt lager. Aktiveringsfunktionen som `nnet` använder är sigmoidfunktionen och kostnadsfunktionen är maximala villkorliga sannolikheten (maximum conditional likelihood). Antalet noder i det dolda lagret och även viktnedbrytningen är valbara parametrar. Värdena på antalet noder som testas är två, fem och tio och för viktnedbrytningen: 0, 10^{-1} och 10^{-4}.

4.2 Utvärdering och jämförelse av klassificeringsmetoder

I detta projekt används nästlad korsvalidering för att jämföra olika klassificeringsmetoders förmåga att skilja på sjuka och friska försökspersoner baserat på mätningar från fot, vad eller både fot och vad. Vi har valt att modifiera den nästlade korsvalideringen genom att tillämpa två olika klassificeringsmått i den inre- och yttre korsvalideringen. Båda dessa mått är baserade på Cohen's kappa, som är lämpligt att använda för våra datamängder (fot, vad samt både fot och vad) eftersom relativt få observationer av sjuka försökspersoner ingår.

Klassificeringsmåttet i den inre korsvalideringen kallar vi "minsta kappa". Detta mått räknas ut genom att identifiera det minsta kappavärdet som erhålls då en modell valideras med de olika valideringsmängderna i den inre korsvalideringen. Modellen som ger högst minsta kappa selekteras. Vi har valt detta mått eftersom vi i tidiga studier av våra modeller observerade att kappavärdet som erhölls per indelning i korsvalideringarna kunde variera kraftigt. Genom att välja modellen med högsta minsta kappa är vår förhoppning att den valda modellen är mer robust när den testas på nya observationer.

I den yttre korsvalideringen kommer vi istället att använda klassificeringsmåttet "totalt kappa". Detta mått beräknas genom att slå samman klassificeringarna av alla testmängder i den yttre korsvalideringen och därefter beräkna kappa. Vi valde att använda detta mått (istället för t.ex. medelvärdet av kappa för testmängderna) för att få ett mer robust klassificeringsmått. Skillnaden mellan detta mått och minsta kappa är att, medan minsta kappa är inriktat på att ge mer stabila modeller, ger totalt kappa en uppfattning om hur väl klassificeringen fungerar.

För träning, validering och testning av modeller inkluderas i första hand endast bekräftat sjuka och friska försökspersoner. Vi kommer dessutom att undersöka effekten av att träna modellerna med 30% av de obekräftat sjuka, som då antas vara sjuka. Validering och testning sker dock endast med bekräftat sjuka och friska försökspersoner.

Uppdelningen av datamängden i inre och yttre indelningar i den nästlade korsvalideringen görs slumpmässigt och stratifierat. Detta säkerställer att andelen observationer från de olika grupperna i varje indelning är ungefär samma som i hela datamängden, vilket är viktigt eftersom det i datamängderna vi studerar är relativt få observationer av bekräftat sjuka jämfört med friska. Uppdelningen görs också så att mätningar från samma person hamnar i samma indelning. Detta görs för att undvika att modellen lär sig att känna igen en viss persons svettmönster. Eftersom

resultatet från den nästlade korsvalideringen kan bero på den specifika uppdelningen av datamängden, testar vi dessutom olika slumpmässiga uppdelningar av datamängden för att undersöka hur robusta våra resultat är.

Samma antal uppdelningar används i yttre och inre korsvalideringen (d.v.s. $m = n$). Då antalet observationer i gruppen med bekräftat sjuka är få i båda datamängderna (18 mätningar på vad, 9 mätningar på fot) väljs antal uppdelningar specifikt per datamängd så att varje valideringss- eller testmängd innehåller minst två observationer från gruppen med bekräftat sjuka. Detta motsvarar $m = n = 5$ för vadmätningar och $m = n = 3$ för fotmätningar.

I den nästlade korsvalideringen standardiseras kovariaterna utifrån träningsmängderna. Standardiseringen görs genom att på varje träningsmängd och per kovariat subtrahera medianen och dela med interkvartilavståndet. Detta säkerställer att mätskalan blir ungefär samma för alla kovariater samt gör deras fördelningar mer symmetriska. Samma standardisering tillämpas sedan på kovariaterna i motsvarande valideringss- eller testmängd.

I den inre korsvalideringen skapas modeller utifrån alla möjliga kombinationer av valbara parametrar som anges för klassificeringsmetoden i avsnitt 6.1. Dessutom testas modellerna med sannolikhetströskelvärden 0.1, 0.2, 0.3, 0.4, 0.5 samt 0.6, där lägre sannolikhetströskelvärden än 0.5 främjar klassificering av observationer som sjuka. Val av parametrar och sannolikhetströskelvärden görs tillsammans med variabelselektion, som antingen sker via stegvis framåtselektion eller stegvis bakåtselektion. Därefter väljs modellen med högsta minsta kappa.

Den valda modellen tränas om på hela träningsmängden i den yttre korsvalideringen och testas med motsvarande testmängd. När alla m modellerna som selekterats i de m inre korsvalideringarna testats med motsvarande testmängd, räknas kappa per modell samt totalt kappa ut. För mer information om den algoritm som används för att utvärdera klassificeringsmetoderna, se bilaga 5.

Slutligen väljs per mätområde den klassificeringsmetod som uppvisar högst totalt kappa som bästa metod.

4.3 Klassificering av obekräftat sjuka

För att i slutändan kunna använda klassificeringen som ett sätt att diagnostisera patienter är det de obekräftat sjuka vi vill undersöka. Denna grupp består eventuellt av både sjuka och friska individer. För att analysera dessa mätningar tränas en modell upp baserad på den bästa klassificeringsmetoden per mätområde. Modellen tränas med alla mätningar från de friska och bekräftat sjuka på det aktuella mätområdet. Selektionen av modellen kan sedan utföras genom korsvalidering motsvarande den inre korsvalideringen som redogörs i avsnitt 4.2. Dessa modeller testas därefter på de obekräftat sjuka på respektive mätområde och antalet friska samt sjuka klassificeringar noteras. Vi jämför sedan klassificeringarna av modellerna tränade på fot respektive vad genom att studera klassificeringen av fot- och vadmätningar utförda på samma försöksperson. Bland de obekräftat sjuka finns det 26 personer som genomgått mätningar på både fot och vad.

5 Resultat

Resultatdelen innehåller i huvudsak två delar. Den första delen handlar om utvärdering och jämförelse av de tre olika klassificeringsmetoderna. Den andra delen undersöker hur de bästa modellerna per mätområde klassificerar de obekräftat sjuka. Endast de bästa klassificeringsmetoder och modeller per mätområde med tillhörande förvirringsmatris redovisas nedan. Alla andra körningar kan ses i bilagorna 7-9.

Tabellerna i bilagorna innehåller också körningar där 30% av andelen obekräftat sjuka är inkluderade, dessa har vi inte valt att redovisa i tabellform här. En anledning till detta är dilemmat med att använda en modell tränad på obekräftat sjuka för klassificering av obekräftat sjuka. Detta går

att komma runt genom att exkludera de försökspersonerna som används till träning för att sedan klassificera obekräftat sjuka. Men då minskas mängden av obekräftat sjuka att klassificera.

5.1 Utvärdering och jämförelse av klassificeringsmetoder

I tabell 2 redovisas resultatet för de bästa klassificeringsmetoderna per mätområde. Slumpmässig skog ger bäst totalt kappa för vadmätningarna medan neurala nätverk ger bäst totalt kappa på de resterande två mätområdena. Högst totalt kappa fås för vadmätningarna medan lägst totalt kappa fås för fotmätningarna. Även om mer data är tillgängligt då fot- och vadmätningarna slås ihop verkar detta inte gynna klassificeringen.

Förvirringsmatriserna som ses i figur 6 visar resultatet av klassificeringen för varje klassificeringsmetod. Måttet på totalt kappa är baserat på dessa förvirringsmatriser. En relativt hög andel av de bekräftat sjuka klassificeras som friska. För fotmätningar och vadmätningar så klassificeras 33 % av de bekräftat sjuka som friska, och för fot- och vadmätningar kombinerat så klassificeras 44% av de bekräftat sjuka som friska. Alla klassificeringsmetoder lyckas bra med att klassificera friska om man ser till andelen felklassificeringar.

Tabell 2: Bästa klassificeringsmetod per mätområde. Mätområde specificerar vart mätningarna är tagna ifrån, fot & vad betyder att data från fot och vad är kombinerat.

Mätområde	**Klassificerings-metod**	**Sökmetod**	**Totalt kappa** (högsta minsta kappa användes i inre korsvalideringen)
Vad	Slumpmässig skog	Stegvis-bakåtselektion	**0.6275**
Fot	Neurala nätverk	Stegvis-bakåtselektion	**0.5414**
Fot & vad	Neurala nätverk	Stegvis-framåtselektion	**0.5959**

		Referens: Frisk	Referens: Sjuk
Prediktion	Frisk	147	6
	Sjuk	6	12

a) **Vadmätningar**.

		Referens: Frisk	Referens: Sjuk
Prediktion	Frisk	142	3
	Sjuk	6	6

b) **Fotmätningar**.

		Referens: Frisk	Referens: Sjuk
Prediktion	Frisk	295	12
	Sjuk	6	15

c) **Fot- & vadmätningar**.

Figur 6: Förvirringsmatriser för bästa klassificeringsmetod per mätområde.

Om 30% av de obekräftat sjuka inkluderas i träningen av bästa klassificeringsmetod så ses en förbättring för vadmätningar och fot- och vadmätningarna kombinerat. På vadmätningarna är det klassificeringsmetoden neurala nätverk som är bäst, detta med ett kappa på 0.7085. Anledningen till denna förbättring beror på att färre sjuka felklassificeras, här klassificeras 11 % av de bekräftat sjuka som friska gentemot tidigare 33%. För fot- och vadmätningar kombinerat så är det klassificeringsmetoden slumpmässig skog som är bäst, detta med ett kappa på 0.6100. Här klassificeras 37% av de sjuka som friska, en liten förbättring gentemot tidigare klassificeringsmetod vilket felklassificerade 44%.

5.2 Klassificering av obekräftat sjuka

Den bästa modellen per mätområde räknas fram baserat på den bästa klassifficeringsmetod per mätområde från tabell 2. I tabell 3 redovisas resultatet för bästa modell per mätområde. På vadmätningarna är slumpmässig skog bäst, och den bästa modellen har parametern $mtry = 1$ och ett sannolikhetströskelvärde lika med 0.6. Ett lägre sannolikhetströskelvärde än 0.5 främjar klassificeringen av observationer som sjuka. För fotmätningar och fot- och vadmätningar kombinerat är neurala nätverk bäst. Dock erhålls olika värden på parametrarna samt olika sannolikhetströskelvärden för dessa två modeller. Kovariaten **Hazard_mode_f1** är med i alla de bästa modellerna, vilket kan indikera på att denna kovariat förklarar utfallet bra. Någon generell trend till vilken sökmetod som är mest gynnsam kan inte ses, och mängden kovariater per modell varierar.

Tabell 3: Bästa modell per mätområde.

Mätområde	**Sökmetod**	**Kovariater**	**Parametrar**	**Sannolikhets-tröskelvärde**
Vad	Stegvis-bakåtselektion	Hazard_mode_f1, Hazard_mode_f10, Hazard_mode_f30	mtry = 1	0.6
Fot	Stegvis-bakåtselektion	WAF_f30, Intensity_f10, Avesize_f10, CI300_f1 CI300_f10, Hazard_mode_f1, Hazard_mode_f30	Noder=10 Viktnedbrytning = 10^{-1}	0.2
Fot & vad	Stegvis-framåtselektion	Hazard_mode_f1, WAF_f10, WAF_f1, Avesize_f10, Intensity_f1	Noder=2 Viktnedbrytning = 10^{-1}	0.4

Klassificering av obekräftat sjuka kan göras med de bästa modellerna per mätområde som ses i tabell 3. Resultatet av klassificeringen kan uppenbarligen inte bekräftas eftersom att vi inte vet det sanna tillståndet hos dessa individer. I bilaga 10 kan klassificeringen av obekräftat sjuka för bästa modell per mätområde ses i sin helhet. I tabell 4 är resultatet för klassificering av obekräftat sjuka summerat. För vadmätningarna så klassificeras de flesta obekräftat sjuka individerna som friska medan för fotmätningarna så klassificeras de flesta obekräftat sjuka individer som sjuka. När modellen är tränad på både fot- och vadmätningar så är förhållandet i princip lika, 36 friska vs. 37 sjuka.

Tabell 4: Klassificering av obekräftat sjuka. Tabellen visar hur många individer som klassificeras som sjuka respektive friska. Klassificeringen är gjord med bästa modell per mätområde, vilka ses i tabell 3.

Mätområde	**Frisk**	**Sjuk**	**Totalt**
Vad	38	7	45
Fot	9	19	28
Fot & vad	36	37	73

5.2.1 Jämförelse av modeller för fot och vad

Mätningar från både fot och vad finns tillgängligt för totalt 26 individer. Klassificering på dessa två mätområden kan då jämföras om vi antar en av dessa kategorier som referens. Resultatet kan ses i figur 7, där resultatet från klassificeringen på vadmätningarna är referens. Stor skillnad på klassificeringen kan ses för dessa individer beroende på vilket mätområde datan kommer ifrån. Värdet på kappa är 0.1495, vilket visar på en låg överensstämmelse. Modellen tränad på fotmätningarna klassificerar ett betydligt större antal av individerna som sjuka gentemot modellen tränad på vadmätningarna.

		Referens	
		Frisk	Sjuk
Prediktion	Frisk	8	0
	Sjuk	14	4

Figur 7: Förvirringsmatris för jämförelse av modeller på mätområdena **fot** och **vad**. Klassificering för modellen tränad på vadmätningarna har antagits vara referens.

6 Diskussion

Från resultatet kan vi se att bäst klassificering erhölls med metoden slumpmässig skog från mätningarna gjorda på vad där kovariaterna valdes med stegvis bakåtselektion. Denna metod ger rätt klassificering av 67% av de bekräftat sjuka försökspersonerna och 96% av de friska försökspersonerna. Olika faktorer som kan ha påverkat resultatet diskuteras först baserat på datamängden och hur den blivit insamlad och efter det på hur metoden varit uppbyggd. Tillsist diskuteras framtida forskningsinriktningar.

Den totala mängden data i detta projekt är relativt liten, vilket är ett vanligt problem när det kommer till medicinska studier. Datan innehåller flera avvikande observationer, vilket kan ses i bilaga 6. Avvikande observationer i samband med en begränsad mängd data kan vara ett problem, eftersom de avvikande observationerna får större påverkan än om datamängden hade varit större. Således kan det vara svårare för klassificeringsmetoderna att avläsa mönster i datan.

Det är främst data från fotmätningar som finns i begränsad mängd. Det är en nackdel då sjukdomstillståndet tidigast påvisas kliniskt i fötterna. En anledning till att det har samlats in lite data från detta område kan vara att det är svårt att hitta en tillräckligt stor plan yta på fötter för att ta bilder [14]. Ytterligare en faktor som kan påverka datan är att mätningarna från de olika grupperna är tagna vid olika tidpunkter. Datan från grupperna obekräftat samt bekräftat sjuka är insamlade år 2012, medan mätningarna från kontrollgruppen är tagna huvudsakligen under år 2014. Detta kan leda till variation vilket kan göra det svårare att jämföra grupperna. Variationen kan exempelvis bero på hur kameran har kalibrerats eller att olika personer utfört mätningarna. Detta är något som vi kan ha i åtanke när det gäller jämförelsen av de bekräftat sjuka och friska.

Information kring insamlingen av datan är bristfällig, det är framförallt inklusions- och exklusionskriterier som saknas. Inklusions- och exklusionskriterier vilka kriterier försökspersonen måste respektive inte får uppfylla för att inkluderas i studien. På vilket sätt försökspersonerna har identifierats som friska vet vi inte, och vi vet inte heller åldern eller kön för patienterna från den här gruppen, vilket kan påverka svettsekretionen [14]. Om kontrollpersonerna inte testats för perifer neuropati kan det finnas en möjlighet att de har perifer neuropati utan att veta om det. Om så vore fallet introduceras det brus i kontrollgruppen. Det kan också finnas olika grader av perifer neuropati. Därför skulle det kunna finnas en gråzon där en person varken är frisk eller sjuk, utan istället befinner sig emellan. Detta kan också bidra till brus i datan.

De bekräftat sjuka har genomgått ett test för perifer neuropati men diagnosen kan vara osäker. Vi saknar kunskap om vilken undersökning som utförts för att bekräfta perifer neuropati och var denna mätning har gjorts. Om mätningen gjorts på fot är det möjligt att patienten inte har utvecklat perifer neuropati i vaderna ännu. Mätning med dynamiskt svettest (DST) på vad kan då ge ett resultat som liknar en frisk försöksperson. Vi har dessutom ingen information om tidsintervallet mellan testet som bekräftar neuropati och genomförandet av DST. Hos patienter som genomgått cellgiftsbehandling har det observerats att perifer neuropati kan uppkomma under pågående behandling, men det kan också dröja flera veckor eller månader innan symptomen uppkommer [11]. Däremot fann en metaanalys baserad på 31 studier att närmare 70% av patienterna uppvisade

perifer neuropati en månad efter avslutad behandling, vilket minskade till 30% 2 månader senare [10]. Om det bekräftande testet gjorts långt före DST finns därför en risk att den perifera neuropatin försvunnit vid tidpunkten för mätning med DST, medan om det bekräftande testet gjorts långt efter DST finns det en risk att personen inte utvecklat perifer neuropati vid tidpunkten för mätning med DST. Alla dessa faktorer är möjliga källor till en osäker diagnos av de bekräftat sjuka.

Mätningarna gjorda på vad ger bäst klassificering i vår studie (se tabell 2). Detta beror sannolikt inte enbart på att mer data är tillgänglig gentemot fotmätningarna, eftersom att skillnaden är liten. För vadmätnignarna är det totalt 171 observationer, och för fotmätningarna är det totalt 157 observationer. Den största skillnaden dem emellan är att dubbelt så många observationer av bekräftat sjuka finns tillgängligt för vadmätningarna. Således är 10.5% av vadmätningarna gjorda på bekräftat sjuka respektive 5.7% för fotmätningarna. Därmed har klassificeringsmodeller tränade med vadmätningar större möjlighet att lära sig känna igen hur mätningar från sjuka respektive friska ser ut. Modellerna som tränas på både fot- och vadmätningarna ger däremot en sämre klassificering än fot- och vadmodellernas klassificering separat (se tabell 2). Detta tyder på att mätningarna från fot och vad skiljer sig åt.

Gruppen med obekräftat sjuka innehåller eventuellt en blandning av sjuka och friska. Intressant nog uppvisar denna grupp och gruppen med bekräftat sjuka generellt sett liknande mätvärden (se bilaga 6), det kan tyda på att en hög andel av de obekräftat sjuka faktiskt är sjuka. När vi inkluderar 30% av de obekräftat sjuka i träningen av modellerna blir klassificeringen av de bekräftat sjuka och friska bättre (högre totalt kappa) för vadmätningarna samt för fot- och vadmätningarna kombinerat. För vadmätningarna minskar t.ex. felklassificeringen av sjuka som friska från 33% till 11%.

För fotmätningarna blir klassificeringen dock inte bättre (totalt kappa ökar ej) då 30% av de obekräftat sjuka används vid träningen av modellerna. En hypotes som kan förklara våra resultat är att det i gruppen obekräftat sjuka finns personer som befinner sig i ett tidigt skede i sjukdomsförloppet. Fotmätningarna från de obekräftat sjuka skulle i så fall likna fotmätningarna från de bekräftat sjuka, sannolikt med mer brus eftersom vissa av försökspersonerna kan vara friska. Detta förklarar den sämre klassificeringen av fotmätningarna då obekräftat sjuka inkluderas vid träning av modellerna. Vadmätningarna hos dessa försökspersoner skulle däremot kunna bestå av två delmängder: en delmängd som liknar vadmätningarna gjorda på bekräftat sjuka samt en delmängd av mätningar som representerar ett tidigt stadie av sjukdomsförloppet. Om förekomsten av den senare delmängden är relativt hög i gruppen av de obekräftat sjuka medan den är lägre i gruppen bekräftat sjuka, skulle de obekräftat sjuka kunna bidra med ny information då modellen tränas. Detta skulle resultera i en modell som blir bättre på att känna igen personer som endast har lindriga nervskador i vaderna vilket skulle förklara varför vadmodellernas klassificeringar gynnas då mätningar från obekräftat sjuka inkluderas.

Samma hypotes skulle kunna förklara resultaten från klassificeringen av de obekräftat sjuka. Denna klassificering gjordes per mätområde med en modell som tränats på alla bekräftat sjuka och friska observationer. Här identifierar modellen tränad med fotmätningar de flesta obekräftat sjuka som sjuka medan modellen tränad på vadmätningar klassificerade de flesta obekräftat sjuka som friska (se tabell 3). Detta är ännu mer uppenbart då modellerna tränade på fot- och vadmätningar jämförs genom att klassificera de 26 obekräftat sjuka försökspersoner som genomgått mätningar på både fot och vad, där 18 personer klassificeras som sjuka baserat på fotmätningarna jämfört med 4 personer baserat på vadmätningarna (se tabell 7). Om det finns en hög andel personer i gruppen med obekräftat sjuka som har perifer neuropati i fötterna men endast tidiga symptom på perifer neuropati i vaderna, skulle detta kunna förklara det observerade resultatet. Detta är speciellt troligt om andelen personer med lindriga symptom i vaderna i gruppen bekräftat sjuka är liten, eftersom modellen som tränats då inte har lärt sig att känna igen sådana patienter.

I den här studien har vi funnit att modellerna som skapas per mätområde skiljer sig åt mycket. Detta gäller t.ex. för vilka kovariater som ingår i modellerna. Intressant är att kovariaten **Hazard_mode_f1** väljs i alla av de bästa modellerna (se tabell 3). För denna kovariat är mätvärdena för bekräftat sjuka generellt högre än för de friska försökspersonerna (se bilaga 6), även

om vissa observationer avviker från detta mönster. Den stora variationen i vilka kovariater som väljs i övrigt kan bero på att de bekräftat sjuka och friska uppvisar mindre skillnad i mätvärden för dessa kovariater.

Antalet kovariater samt vilka kovariater som ska ingå i modellen väljs med hjälp av stegvis bakåt- eller framåtselektion. Därmed söks inte alla kombinationer av kovariater igenom. Det vi får fram är ett lokalt maximum istället för ett globalt maximum. Möjligtvis kan det alltså finnas någon kombination av kovariater som ger ett bättre resultat, men det har inte undersökts i det här projektet. Generellt väljs en mindre mängd kovariater då stegvis framåtselektion används jämfört med stegvis bakåtselektion. En mindre mängd kovariater är att föredra så länge prestationen av modellen inte minskas alltför drastiskt. I bilagorna 7-9 kan vi se att stegvis bakåtselektion genererar i snitt ett högre värde på kappa. Det finns dock fall då stegvis framåtselektion är bättre, vilket den är exempelvis för fot- och vadmätningar kombinerat. Totalt kappa som mått tar inte hänsyn till hur många kovariater som inkluderas, utan tar endast hänsyn till hur bra klassifikationen är. Ett mått såsom Akaike Information Criterion (AIC) skulle kunna användas för att straffa inklusionen av många kovariater i modellen. Detta är en vanlig avvägning som måste göras inom modellselektion. I vissa fall kan man vara villig att ge upp en viss grad prestation för enklare tolkning av modellen.

Klassificeringsmetoderna som vi använder i detta projekt kommer färdiga från paketet `caret`. Koden är effektiviserad på ett sådant sätt så att valet av möjliga parametrar per klassificeringsmetod i viss mån är begränsade. Metoderna använder således en uppsättning av standardinställningar. Därav finns det stora möjligheter till att förändra dessa metoder ytterligare. I fallet med k-NN är valet av distansmått förinställt som det euklidiska avståndet. Detta skulle potentiellt gå att ändra till något annat mått, såsom manhattandistansen. För metoden `nnet`, vilket används som vårt neurala nätverk, finns det en rad olika parametrar att undersöka. Kostnadsfunktionen och aktiveringsfunktionen är två fundamentala parametrar som är förbestämda. Istället för sigmoidfunktionen som aktiveringsfunktion så skulle till exempel den linjära rektifierade aktiveringsfunktionen vara ett möjligt val. För slumpmässig skog är bland annat antalet beslutsträd och vilket orenhetsmått förbestämt. Istället för Gini Index som mått på orenhet skulle också ett mått på entropi vara ett möjligt val. Hur alla dessa förändringar hade påverkat klassificeringen i vårt fall är det svårt att uttala sig om. Men det är värt att pointera att det möjligtvis finns mer optimala parametrar för alla klassificeringsmetoder än de vi har använt i detta projekt.

I vidare arbete är det viktigt att diskutera huruvida det är värre att klassificera en frisk patient som sjuk (typ I-fel) eller en sjuk patient som frisk (typ II-fel). Vi har utgått från att båda dessa fel viktas lika. Kappa är ett klassificeringsmått som försöker väga missklassificeringar mot ojämnheten i datan, men tar inte hänsyn till vilket typ av fel som görs. Beroende på den tilltänkta tillämpningen av modellen kan det vara av intresse att minimera typ I- eller typ II-felet. I så fall är kappa inte ett rättvisande mått på hur bra klassificeringen blir. Med mer kunskap om den tilltänkta tillämpningen av dynamiskt svettest skulle kappa kunna viktas i träningen av modellerna så att färre faktiskt sjuka klassificeras som friska, eller att färre friska klassificeras som sjuka. För att avgöra vilket fel som bör prioriteras och hur stort felet maximalt får vara behövs en läkares expertis.

7 Slutsats

Syftet med det här projektet var att undersöka om det är möjligt att bedöma om en patient lider av perifer neuropati eller ej genom att analysera patientens svettmönster på fot eller vad. Resultatet från vår studie visar att, bäst klassificering erhölls med klassificeringsmetoden slumpmässig skog på mätningar från vad där kovariaterna väljs med stegvis bakåtselektion. Denna metod ger rätt klassificering av 67% av de bekräftat sjuka försökspersonerna och 96% av de friska försökspersonerna. Vidare har vi identifierat att kovariaten **Hazard_mode_f1**, som beskriver tomrummet mellan svettfläckar en sekund efter att mätningen påbörjats, verkar innehålla mycket information som är relevant för en god klassificering. I framtida arbete bör implementationen av ett viktat kappa övervägas för att minimera typ I- eller typ II-felet som modellen ger.

Referenser

[1] P. Kashyap, *Industrial Applications of Machine Learning. In: Machine Learning for Decision Makers*. Apress, 2017.

[2] J. C. Watson och P. J. B. Dyck, "Peripheral neuropathy: A practical approach to diagnosis and symptom management", *Mayo Clin Proc*, 2015.

[3] T. Ansved. (jan. 2017). Autonoma perifera neuropatier, URL: `https://medibas.se/handboken/kliniska-kapitel/neurologi/tillstand-och-sjukdomar/neuropatier/perifera-neuropatier/` (hämtad 2019-02-01).

[4] P. A. Low, P. E. Caskey, R. R. Tuck, R. D. Fealey och P. J. Dyck, "Quantitative sudomotor axon reflex test in normal and neuropathic subjects", *Ann Neurol*, 1983.

[5] L. Sullivan. (). Nonparametric tests, URL: `http://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704_nonparametric/BS704_Nonparametric_print.html` (hämtad 2019-03-12).

[6] S. Russell och P. Norvig, *Artificial Intelligence: Pearson New International Edition: A Modern Approach*, 3. utg. Prentice Hall, 2013.

[7] T. Ansved. (febr. 2013). Autonoma perifera neuropatier, URL: `https://medibas.se/handboken/kliniska-kapitel/neurologi/tillstand-och-sjukdomar/neuropatier/autonoma-perifera-neuropatier/` (hämtad 2019-02-01).

[8] B. Eliasson. (aug. 2016). Diabetesneuropati, URL: `https://medibas.se/handboken/kliniska-kapitel/neurologi/tillstand-och-sjukdomar/neuropatier/diabetesneuropati/#Screening` (hämtad 2019-02-01).

[9] Z. Iqbal, S. Azmi, R. Yadav, M. Ferdousi, M. Kumar, D. J. Cuthbertson, J. Lim, R. A. Malik och U. Alam, "Diabetic peripheral neuropathy: Epidemiology, diagnosis, and pharmacology", *Clinical Therapeutics*, 2018.

[10] M. Seretny, G. L. Currie, E. S. Sena, S. Ramnarine, R. Grant, M. R. MacLeod, L. A. Colvin och M. Fallon, "Incidence, prevalence, and predictors of chemotherapy-induced peripheral neuropathy: A systematic review and meta-analysis", *Pain*, årg. 155, nr 12, 2014.

[11] R. Zajączkowska, M. Kocot-Kępska, W. Leppert, A. Wrzosek, J. Mika och J. Wordliczek, "Mechanisms of chemotherapy-induced peripheral neuropathy", *Int J Mol Sci*, årg. 20, nr 6, 2019.

[12] N. P. Staff, A. Grisold, W. Grisold och A. J. Windebank, "Chemotherapy-induced peripheral neuropathy: A current review", *Ann Neurol*, årg. 81, nr 6, 2017.

[13] S. Attvall. (febr. 2019). Diabetesfoten, URL: `https://www.internetmedicin.se/page.aspx?id=1531` (hämtad 2019-02-01).

[14] A. Loavenbruck, N. Sit, V. Provitera och W. Kennedy, "High-resolution axon reflex sweat testing for diagnosis of neuropathy", *Clin Auton Res*, 2018.

[15] V. Provitera, M. Nolano, G. Caporaso, A. Stancanelli, L. Santoro och W. R. Kennedy, "Evaluation of sudomotor function in diabetes using the dynamic sweat test", *Neurology*, 2010.

[16] T. Hastie, R. Tibshirani och J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2017.

[17] G. C. Cawley och N. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation", *Journal of Machine Learning Research*, 2010.

[18] T. Srivastava. (mars 2018). Introduction to k-nearest neighbors: Simplified (with implementation in python), URL: `https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/` (hämtad 2019-02-07).

[19] B. Klein. (2018). K-nearest-neighbor classifier, URL: `https://www.python-course.eu/k_nearest_neighbor_classifier.php` (hämtad 2019-02-07).

[20] S. Fortmann-Rose. (juni 2012). Understanding the bias-variance tradeoff, URL: `http://scott.fortmann-roe.com/docs/BiasVariance.html` (hämtad 2019-03-27).

[21] R. Adams och C.Essex, *Calculus: A Complete Course*. Pearson, 8th edition, 2013.

[22] E. Alpaydın. (2010). Introduction to Machine Learning Second Edition, URL: https://kkpatel7.files.wordpress.com/2015/04/alppaydin_machinelearning_2010.pdf.

[23] W.-Y. Loh, "Fifty years of classification and regression trees (with discussion)", *International Statistical Review*, årg. 34, s. 329–370, 2014. DOI: http://www.stat.wisc.edu/~loh/treeprogs/guide/LohISI14.pdf.

[24] W.-Y. Loh. (febr. 2011). Classification and regression trees, URL: https://www.stat.wisc.edu/~loh/treeprogs/guide/wires11.pdf (hämtad 2019-02-16).

[25] N. Donges. (febr. 2018). The random forest algorithm, URL: https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd (hämtad 2019-02-10).

[26] Michael A. Nielsen. (). A beginner's guide to neural networks and deep learning, URL: https://skymind.ai/wiki/neural-network (hämtad 2019-03-06).

[27] Luke Dormehl. (jan. 2019). What is an artificial neural network? here's everything you need to know, URL: https://www.digitaltrends.com/cool-tech/what-is-an-artificial-neural-network/ (hämtad 2019-03-06).

[28] Michael A. Nielsen. (okt. 2018). Using neural nets to recognize handwritten digits, URL: http://neuralnetworksanddeeplearning.com/chap1.html (hämtad 2019-03-07).

[29] O. Omidvar och D. L. Elliott, *Neural Systems for Control*. springer, 2017. DOI: DOI10.1016/B978-012526430-3/50001-5. URL: http://www-bcf.usc.edu/~gareth/ISL/ISLR%5C%20Seventh%5C%20Printing.pdf.

[30] G. James, D. Witten, T. Hastie och R. Tibshirani, *An Introduction to Statistical Learning*. Academic Press, 1997. DOI: DOI10.1007/978-1-4614-7138-7. URL: https://www.sciencedirect.com/topics/computer-science/activation-function.

[31] D. Krstajic, L. J. Buturovic, D. E. Leahy och S. Thomas, "Cross-validation pitfalls when selecting and assessing regression and classification models", *Journal of Cheminformatics*, 2014.

[32] J. Wainer och G. C. Cawley, "Nested cross-validation when selecting classifiers is overzealous for most practical applications", *Journal of Machine Learning Research*, 2017.

[33] Statens beredning för medicinsk och social utvärdering. (2017). Utvärdering av metoder i hälso- och sjukvården och insatser i socialtjänsten, URL: https://www.sbu.se/contentassets/d12fd955318f4feab3709d7ebcc9a72b/sbushandbok.pdf (hämtad 2019-05-09).

[34] L. A. Jeni, J. F. Cohn och F. D. L. Torre, "Facing imbalanced data–recommendations for the use of performance metrics", *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, 2013. DOI: 10.1109/acii.2013.47.

[35] R Core Team, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, 2017. URL: https://www.R-project.org/.

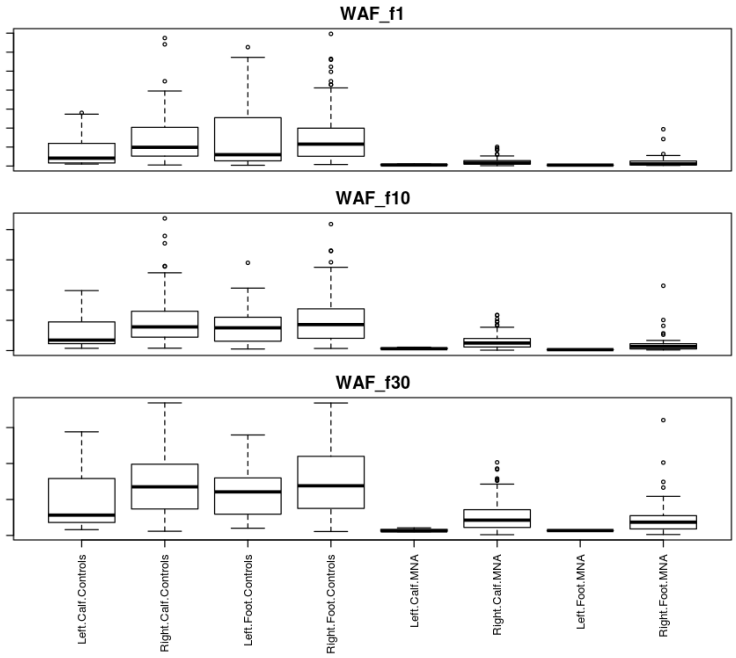
[36] M. Kuhn, *Caret: Classification and regression training*, R package version 6.0-81, 2018. URL: https://CRAN.R-project.org/package=caret.

Bilagor

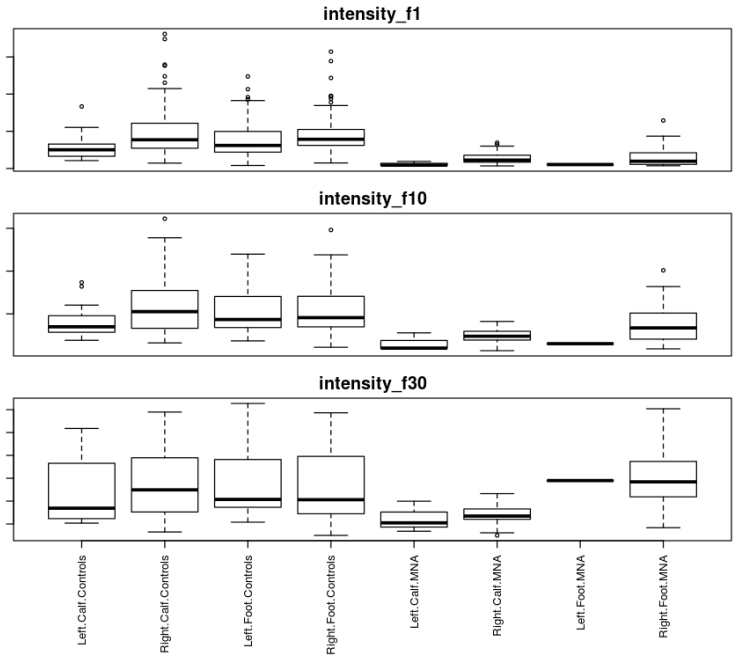
Bilaga 1 : Jämförelse av mätningar med avseende på kroppshalva

De flesta av mätningarna i hela datamängden är gjorda på höger kroppshalva (353 jämfört med 48 på vänster sida). Av vadmätningarna på vänster kroppshalva är 22 friska försökspersoner, 2 obekräftat sjuka och 1 bekräftat sjuk. Av fotmätningarna på vänster kroppshalva är 22 friska försökspersoner och 1 obekräftat sjuk.

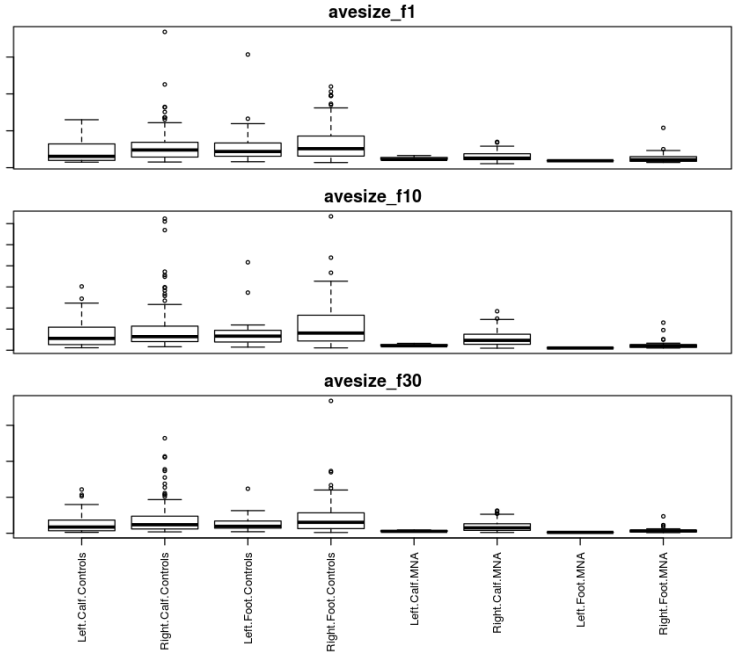
I figur 1-5 ses att de friska försökspersonerna uppvisar liknande värden på höger och vänster kroppshalva med avseende på de 15 kovariaterna. Detta är inte lika tydligt i den s.k. MNA gruppen som innehåller bekräftat och obekräftat sjuka. En anledning till detta kan vara att mycket få mätningar gjorts på vänster kroppshalva (3 vad, 1 fot). Däremot avviker dessa mätningar inte avsevärt jämfört med mätningarna gjorda på höger kroppshalva.



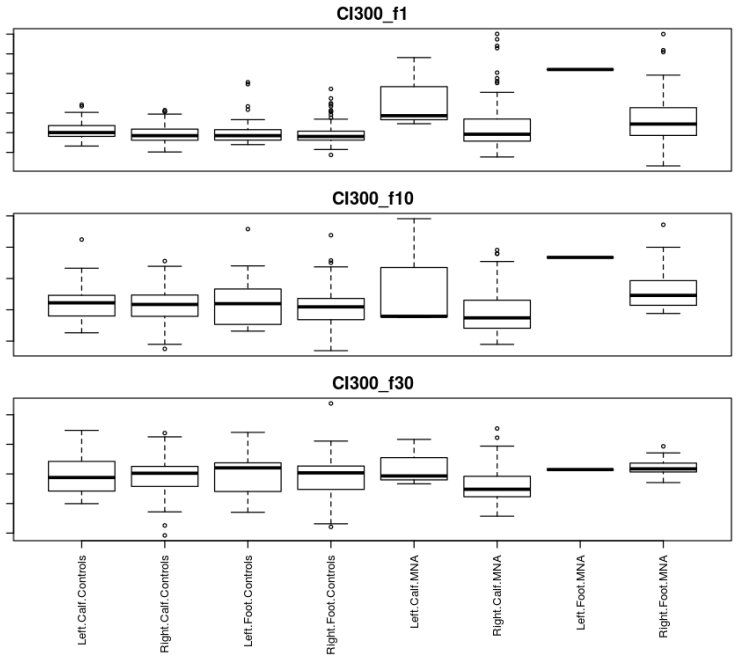
Figur 8: Låddiagram för kovariaterna **WAF_f1** (överst), **WAF_f10** (mitten) och **WAF_f30** (nederst) uppdelade på kroppshalva (left eller right), mätområde (calf eller foot) samt grupper av försökspersoner som här består av Controls (friska försökspersoner) och MNA (bekräftat sjuka och obekräftat sjuka).



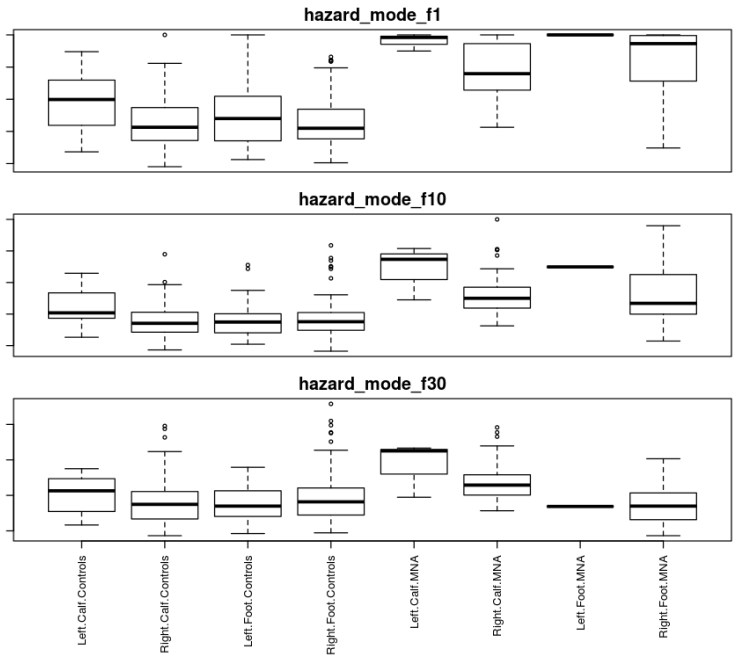
Figur 9: Låddiagram för kovariaterna **intensity_f1** (överst), **intensity_f10** (mitten) och **intensity_f30** (nederst) uppdelade på kroppshalva (left eller right), mätområde (calf eller foot) samt grupper av försökspersoner som här består av Controls (friska försökspersoner) och MNA (bekräftat sjuka och obekräftat sjuka).



Figur 10: Låddiagram för kovariaterna **avesize_f1** (överst), **avesize_f10** (mitten) och **avesize_f30** (nederst) uppdelade på kroppshalva (left eller right), mätområde (calf eller foot) samt grupper av försökspersoner som här består av Controls (friska försökspersoner) och MNA (bekräftat sjuka och obekräftat sjuka).



Figur 11: Låddiagram för kovariaterna **CI300_f1** (överst), **CI300_f10** (mitten) och **CI300_f30** (nederst) uppdelade på kroppshalva (left eller right), mätområde (calf eller foot) samt grupper av försökspersoner som här består av Controls (friska försökspersoner) och MNA (bekräftat sjuka och obekräftat sjuka).



Figur 12: Låddiagram för kovariaterna **hazard_mode_f1** (överst), **hazard_mode_f10** (mitten) och **hazard_mode_f30** (nederst) uppdelade på kroppshalva (left eller right), mätområde (calf eller foot) samt grupper av försökspersoner som här består av Controls (friska försökspersoner) och MNA (bekräftat sjuka och obekräftat sjuka).

Bilaga 2 : Totalt antal mätningar per mätområde och per grupp av försökspersoner samt antal mätningar per grupp av försökspersoner

Det totala antalet mätningar per mätområde och per grupp av försökspersoner sammanfattas i tabell 5 medan antal mätningar per försöksperson i de olika grupperna av försökspersoner redovisas i tabell 6. Noterbart i tabell 6 är att sju försökspersoner (alla friska) genomgick fler än två mätningar. På dessa sju försökspersoner har mer än en mätning gjorts per mätområde.

Tabell 5: Totalt antal mätningar per mätområde och per grupp av försökspersoner.

Mätområde	Fot	Vad
Antal mätningar		
Totalt	185	216
Bekräftat sjuka	9 (4,86%)	18 (8,33%)
Obekräftat sjuka	28 (15,14%)	45 (20,83%)
Friska	148 (80,00%)	153 (70,83%)

Tabell 6: Antal mätningar per försöksperson i de olika grupperna av försökspersoner.

Grupp	**Antal mätningar**	**Antal försökspersoner**	**Mätområde(n)**
Bekräftat sjuka	1	9	Vad
	2	9	Fot och vad
Obekräftat sjuka	1	21	Fot eller vad
	2	26	Fot och vad
Friska	1	20	Fot eller vad
	2	93	Fot och vad
	6, 10, 13, 14, 16, 17 eller 19	1	Fot och vad

Bilaga 3 : Beskrivning av variablerna i datamängden förutom kovariaterna

Variablerna som ingår i datamängden, förutom kovariaterna, beskrivs kortfattat i tabell 7.

Tabell 7: Beskrivning av variablerna i datamängden förutom kovariaterna.

Variabelnamn	**Beskrivning**
id	Försökspersonens unika identifieringsnummer
location	Mätområde: fot eller vad
side	På vilken kroppshalva mätningen gjordes: höger eller vänster
date	Datum, ÅÅÅÅ-MM-DD
time	Tid, HH-MM-SS, AM eller PM
condition	Controls (friska kontroller) eller MNA (obekräftat eller bekräftat sjuka)
did	Unikt identifieringsnummer per mätning
condition_extra	neuropathic (bekräftat sjuka) eller unknown (friska kontroller eller obekräftat sjuka)

Bilaga 4 : R-kod

```
### Innehåll ##################################################
# A----DATA & R-PAKET
# A-01 Import av data, skapa variabel 'label', uppdelning
#      i fot- och vadmätningar, vektor med kovariatnamn
# A-02 R-paket
# B----FUNKTIONER
# B-01 step_median_center
# B-02 iqr_func
# B-03 step_iqr_scale
# B-04 create_inner_folds
# B-05 create_recipe
# B-06 grid_search
# B-07 inner_cv
# B-08 class_converter
# B-09 outer_cv
# C---- IDENTIFIERING AV BÄSTA METOD PER MÄTOMRÅDE - EXEMPEL
# C-01 knn
# C-02 rf
# D --- KLASSIFICERING AV OBEKRÄFTAT SJUKA - EXEMPEL
# D-01 Bästa metod på fot
# D-02 Bästa metod på vad
# D-03 Jämförelse av modeller för fot och vad
# D-04 Bästa metod på all data
###############################################################

# Koden är kompatibel med:
#   - R version 3.4.2
#   - caret version 6.0-81
#   - recipes version 0.1.4
#   - groupdata2 version 1.0.0

### A-01 Import av data, etc ##################################

# Importera data
df=read.csv('spots_example_df_3frames_v2.txt',skip=15)

# Kategori sjuk är MNA
df$condition=factor(df$condition, levels = c('MNA','Controls'))

# skapa label: control, neuropathic, unknown
df$label=rep(0,dim(df)[1])
df[df$condition=='Controls',]$label='control'
df[df$condition=='MNA' & df$condition_extra=='unknown',]$label='unknown'
df[df$condition=='MNA' & df$condition_extra=='neuropathic',]$label='neuropathic'

# Dela upp i fot och vad
foot=df[df$location=='Foot',]
row.names(foot)=seq(1,dim(foot)[1]) # uppdatera radindex
calf=df[df$location=='Calf',]
row.names(calf)=seq(1,dim(calf)[1]) # uppdatera radindex

covariates=c('WAF_f1', 'WAF_f10', 'WAF_f30', 'intensity_f1', 'intensity_f10',
             'intensity_f30', 'avesize_f1', 'avesize_f10', 'avesize_f30',
             'CI300_f1', 'CI300_f10', 'CI300_f30', 'hazard_mode_f1',
             'hazard_mode_f10', 'hazard_mode_f30')

###############################################################

### A-02 R-paket ##############################################

library(caret)
library(recipes)
library(groupdata2)

###############################################################


### B-01 step_median_center ###################################
```

```
69
70 # baserad på: https://github.com/tidymodels/recipes/blob/master/R/center.R
71
72 step_median_center <-
73   function(recipe,
74            ...,
75            role = NA,
76            trained = FALSE,
77            medians = NULL,
78            na_rm = TRUE,
79            skip = FALSE,
80            id = rand_id("median_center")) {
81     add_step(
82       recipe,
83       step_median_center_new(
84         terms = ellipse_check(...),
85         trained = trained,
86         role = role,
87         medians = medians,
88         na_rm = na_rm,
89         skip = skip,
90         id = id
91       )
92     )
93   }
94
95 ## Initializes a new object
96 step_median_center_new <-
97   function(terms, role, trained, medians, na_rm, skip, id) {
98     step(
99       subclass = "median_center",
100      terms = terms,
101      role = role,
102      trained = trained,
103      medians = medians,
104      na_rm = na_rm,
105      skip = skip,
106      id = id
107    )
108  }
109
110 prep.step_median_center <- function(x, training, info = NULL, ...) {
111   col_names <- terms_select(x$terms, info = info)
112   check_type(training[, col_names])
113
114   medians <-
115     vapply(training[, col_names], median, c(median = 0), na.rm = x$na_rm)
116   step_median_center_new(
117     terms = x$terms,
118     role = x$role,
119     trained = TRUE,
120     medians = medians,
121     na_rm = x$na_rm,
122     skip = x$skip,
123     id = x$id
124   )
125 }
126
127 bake.step_median_center <- function(object, new_data, ...) {
128   res <-
129     sweep(as.matrix(new_data[, names(object$medians)]), 2, object$medians, "-")
130   if (is.matrix(res) && ncol(res) == 1)
131     res <- res[, 1]
132   new_data[, names(object$medians)] <- res
133   as_tibble(new_data)
134 }
135
136 print.step_median_center <-
137   function(x, width = max(20, options()$width - 30), ...) {
138     cat("Median centering for ", sep = "")
```

```
    printer(names(x$medians), x$terms, x$trained, width = width)
    invisible(x)
  }


#' @rdname step_center
#' @param x A `step_center` object.
#' @export
tidy.step_median_center <- function(x, ...) {
  if (is_trained(x)) {
    res <- tibble(terms = names(x$medians),
                  value = x$medians)
  } else {
    term_names <- sel2char(x$terms)
    res <- tibble(terms = term_names,
                  value = na_dbl)
  }
  res$id <- x$id
  res
}

######################################################################

### B-02 iqr_func ##################################################

# iqr beräkning
iqr_func=function(col){quantile(col,0.75,names=FALSE)-quantile(col,0.25,names=FALSE
    )}

######################################################################

### B-03 step_iqr_scale ############################################

# baserad på: https://github.com/tidymodels/recipes/blob/master/R/scale.R

step_iqr_scale <-
  function(recipe,
           ...,
           role = NA,
           trained = FALSE,
           iqrs = NULL,
           skip = FALSE,
           id = rand_id("iqr_scale")) {
    add_step(
      recipe,
      step_iqr_scale_new(
        terms = ellipse_check(...),
        role = role,
        trained = trained,
        iqrs = iqrs,
        skip = skip,
        id = id
      )
    )
  }

step_iqr_scale_new <-
  function(terms, role, trained, iqrs, skip, id) {
    step(
      subclass = "iqr_scale",
      terms = terms,
      role = role,
      trained = trained,
      iqrs = iqrs,
      skip = skip,
      id = id
    )
  }

#' @importFrom stats sd
```

```
#' @export
prep.step_iqr_scale <- function(x, training, info = NULL, ...) {
  col_names <- terms_select(x$terms, info = info)
  check_type(training[, col_names])

  iqrs <-
    vapply(training[, col_names], iqr_func, c(iqr_func = 0))
  step_iqr_scale_new(
    terms = x$terms,
    role = x$role,
    trained = TRUE,
    iqrs,
    skip = x$skip,
    id = x$id
  )
}

#' @export
bake.step_iqr_scale <- function(object, new_data, ...) {
  res <-
    sweep(as.matrix(new_data[, names(object$iqrs)]), 2, object$iqrs, "/")
  if (is.matrix(res) && ncol(res) == 1)
    res <- res[, 1]
  new_data[, names(object$iqrs)] <- res
  as_tibble(new_data)
}

print.step_iqr_scale <-
  function(x, width = max(20, options()$width - 30), ...) {
    cat("IQR scaling for ", sep = "")
    printer(names(x$iqrs), x$terms, x$trained, width = width)
    invisible(x)
  }


#' @rdname step_scale
#' @param x A `step_scale` object.
#' @export
tidy.step_iqr_scale <- function(x, ...) {
  if (is_trained(x)) {
    res <- tibble(terms = names(x$iqrs),
                  value = x$iqrs)
  } else {
    term_names <- sel2char(x$terms)
    res <- tibble(terms = term_names,
                  value = na_dbl)
  }
  res$id <- x$id
  res
}

######################################################################

### B-04 create_inner_folds ###################################

create_inner_folds=function(fold,num_inner_folds,i){
  #-------------------------------------------------------------
  # --Delar upp i n inre valideringsfolds - stratifierat
  # fold: data från ett yttre träningsfold
  # num_inner_folds: antal inre folds
  # i: seed
  # RETURNERAR:
  # $inner_train_folds: lista med index för n träningsfolds
  # $inner_val_folds: lista med index för motsvarande testfolds
  #-------------------------------------------------------------

  set.seed(i)
  fold_fold=fold(fold,k=num_inner_folds,cat_col='label',id_col='id') # groupdata2
    paketet
  # returnerar dataframe med foldnummer i variabeln .folds
```

```
277   # Måste omvandlas till lista av vektorer med radindex för att funka med
        trainControl()
278   # skapa tom lista för vektorer med radindex per fold
279   inner_val_folds = vector("list",num_inner_folds)
280   for(i in 1:num_inner_folds){
281     inner_val_folds[[i]]=which(fold_fold$.folds == i) # vektor med radindex för
        fold i
282   }
283   # --
284
285   # -- Dela upp i n inre träningsfolds
286   # skapa tom lista för vektorer med radindex per fold
287   inner_train_folds=vector("list",num_inner_folds)
288   all_idx=seq(1,dim(fold)[1]) # alla radindex i fold
289   for (i in 1:length(inner_val_folds)){
290     inner_train_folds[[i]]=setdiff(all_idx,inner_val_folds[[i]]) # radindexen som
        skiljer
291   }
292   # --
293
294   # -- Exkludera unknown i valideringsfolds
295   temp_val_folds=list()
296   idx=0
297   for (val_fold_idx in inner_val_folds){
298     idx=idx+1
299     val_set=fold[val_fold_idx,]
300     # index som ej har label unknown
301     val_set=val_set[val_set$label!='unknown',] # exkludera unknown
302     temp_val_folds[[idx]]=as.integer(row.names(val_set))
303   }
304   inner_val_folds=temp_val_folds # innehåller endast control & neuropathic
305   # --
306   return(list(inner_train_folds,inner_val_folds))
307 }
308
309 ##################################################################
310
311 ### B-05 create_recipe ###################################
312
313 create_recipe=function(fold,covar_sel){
314   #-----------------------------------------------------------
315   # Skapar ett recept för standardisering av träningsdata:
316   # För varje kovariat subtraheras medianen från värdet följt av delning med
        interkvarilavståndet
317   # Samma standisering tillämpas sedan automatiskt på testdata i train()
318   # OBS: bör alltid köras innan grid_search()
319   # fold: data från ett yttre träningsfold
320   # covar_sel: ett urval av kovariater
321   # RETURNERAR:
322   # $df: data med utvalda kovariater och 'condition'
323   # $recipe: recept (används i train() )
324   #-----------------------------------------------------------
325
326   fold2=fold[,c(covar_sel,'condition')]
327   # -- Skapa RECIPE
328   fold_recipe=recipe(condition ~ .,data=fold2) %>% # condition EJ label
329     ## Subtrahera medianen
330     step_median_center(all_predictors()) %>%
331     ## Dela med IQR
332     step_iqr_scale(all_predictors())
333   # --
334   return(list(df=fold2,recipe=fold_recipe))
335 }
336
337 ##################################################################
338
339 ### B-06 grid_search #####################################
340
341 grid_search=function(method,params,inner_train_folds,inner_val_folds,
342                      fold2,fold_recipe,i,thresholds){
```

```
343 #-----------------------------------------------------------------
344 # grid search (inkl. standardisering av datan, uppdelning i folds)
345 # hittar bästa sannolikhetströskelvärdet
346 # method: namn på method i caret::train
347 # params: hyperparametrar som 'grid'
348 # inner_train_folds: lista med vektorer av radindex för träning
349 # inner_val_folds: lista med vektror av radindex för validering
350 # fold2: data med utvalda kovariater och 'condition'
351 # fold_recipe: recept, används för standardisering av valideringsdata
352 # baserat på träningsdata i train()
353 # i: seed
354 # thresholds: vektor med sannolikhetströskelvärden
355 # RETURNERAR:
356 # $mod: bästa modellen
357 # $threshold: bästa sannolikhetströskelvärde
358 # $kappa: högsta minsta kappa
359 #-----------------------------------------------------------------
360
361 set.seed(i)
362 # -- GRID SEARCH
363 ctrl=trainControl(index=inner_train_folds,
364                   indexOut=inner_val_folds,
365                   savePredictions='all',
366                   classProbs=TRUE)
367 fit=train(x=fold_recipe,data=fold2,method=method,
368           trControl=ctrl,metric='Kappa',tuneGrid=params)
369
370 temp=fit$pred # dataframe med sannolikheter, observerade värden, hyperparametervä
    rden,etc.
371
372 # -- Beräkna klassificering för olika sannolikhetströskelvärden
373 for (t in 1:length(thresholds)){
374   predictions=sapply(fit$pred$MNA,class_converter,threshold=thresholds[t])
375   temp=cbind(temp,predictions)
376   names(temp)[ncol(temp)]=paste('thres',t,sep='_')
377 }
378
379 # -- Hitta största totala kappa, bästa sannolikhetströskelvärdet, bästa
    hyperparametrar
380
381 # initiera tom matris för att spara bästa totala kappavärdene per kombination
    kovariater/sannolikhetströskelvärden
382 tot_kappas=matrix(rep(0,dim(params)[1]*length(thresholds)),nrow=dim(params)[1],
    ncol=length(thresholds))
383
384 # initiera tom matris som sparar minsta kappavärden per kombination kovariater/
    sannolikhetströskelvärden
385 min_kappas=matrix(rep(0,dim(params)[1]*length(thresholds)),nrow=dim(params)[1],
    ncol=length(thresholds))
386
387 # initiera tom vektor för att spara kappa per fold
388 fold_kappas=rep(0,num_inner_folds)
389
390 for (row in 1:dim(params)[1]){
391   param_row=as.data.frame(params[row,])
392   names(param_row)=names(params)
393   temp2=merge(temp, param_row)
394   for (thres in 1:length(thresholds)){
395     col=paste('thres',thres,sep='_') # kolonn för sannolikhetströskelvärdet
396     # Beräkning av totalt kappa
397     pred=temp2[,col] # modellens klassificering
398     obs=temp2$obs # verklig klassificering
399     tot_kappas[row,thres]=as.numeric(defaultSummary(data.frame(obs=obs,pred=pred)
    )[2]) # totalt kappa
400     # Beräkning av kappa per fold
401     resample_levels=unique(as.factor(temp2$Resample))
402     fold_kappas=rep(0,num_inner_folds) # reset
403     for (l in 1:length(resample_levels)){
404       resample=temp2$Resample==resample_levels[l]
405       sel=temp2[resample,]
```

```
406         pred_resample=sel[,col] # modellens klassificering i folden
407         obs_resample=sel$obs # verklig klassificering i folden
408         fold_kappas[1]=as.numeric(defaultSummary(data.frame(obs=obs_resample,pred=
      pred_resample))[2])
409       }
410       min_kappa=min(fold_kappas)
411       min_kappas[row,thres]=min_kappa # spara minsta kappa
412     }
413   }
414   best_tot_kappa=max(tot_kappas) # största totala kappa
415   best_min_kappa=max(min_kappas) # största minsta kappa
416   idx_max=which(min_kappas == best_min_kappa, arr.ind = TRUE) # hitta rad &
      kolonnindex för största minsta kappa
417   min_thres=min(idx_max[,2]) # index för minsta sannolikhetströskelvärdet
418   sel_threshold=thresholds[min_thres]
419   idx_max=idx_max[idx_max[,2]==min_thres,] # Selektera minsta sannolikhetströskelvä
      rdet
420   idx_max=as.data.frame(idx_max)
421   sel_params=params[as.numeric(idx_max[1,1]),] # välj första bästa som bästa
      hyperparametrar
422   sel_params=as.data.frame(sel_params)
423   names(sel_params)=names(params)
424
425   # skapa modell med bästa parametrarna
426   ctrl=trainControl(index=inner_train_folds,
427                     indexOut=inner_val_folds,
428                     classProbs=TRUE)
429   fit=train(x=fold_recipe,data=fold2,method=method,
430             trControl=ctrl,metric='Kappa',tuneGrid=sel_params) # sel_params
431   # --
432   # returnerar modell, threshold, bästa kappa
433   return(list(mod=fit,threshold=sel_threshold,kappa=best_min_kappa))
434 }
435
436 ######################################################################
437
438 ### B-07 inner_cv ##############################################
439
440 inner_cv=function(method,params,covars,fold,i,num_inner_folds,backward,thresholds){
441   #-------------------------------------------------------------
442   # inre korsvalidering
443   # method: namn på method i caret::train
444   # params: hyperparametrar som kolonner i dataframe
445   # kolonnens namn ska motsvara parameterns namn i method
446   # covars: möjliga kovariater som kan väljas
447   # fold: data från yttre träningsfold
448   # i: seed
449   # num_inner_folds: antal inre folds
450   # backward: FALSE: använd forward selection, TRUE: använd backward selection
451   # threshold: vektor med sannolikhetströskelvärden
452   # RETURNERAR:
453   # $best_model: bästa modellen (med $mod=modellen, $threshold=sannolikhetströskelv
      ärde,
454   #              $kappa=bästa kappa från inre korsvalideringen - OBS: används ej i
      yttre korsvalideringen)
455   # $best_covars: kovariaterna som används i bästa modellen
456   # $params: parametrarna som användes, behövs om method='rf'
457   #-------------------------------------------------------------
458
459   # -- Skapa inre folds
460   inner_folds=create_inner_folds(fold,num_inner_folds,i)
461   inner_train_folds=inner_folds[[1]]
462   inner_val_folds=inner_folds[[2]]
463   # --
464
465   # --- Variabelselektion & val av hyperparametrar
466   best_performance=-1
467   best_covars=NULL
468   best_model=NULL
469   num_covars=length(covars) # antal ursprungliga kovariater
```

```
covar_names=covars # kopia

# --- Om method=rf, maximal andel kovariater som slumpas fram i varje nod
prop_mtry=0.4

if (backward==TRUE){
  # --- BACKWARD STEPWISE SELECTION ---
  # ----- Skapa modell med alla kovariater ----------------------
  df_recipe=create_recipe(fold,covars) # covars
  fold2=df_recipe$df # data med utvalda kovariater
  fold_recipe=df_recipe$recipe # för standardisering av data baserat på träningsfold
  # -- Om method=rf, bestäm parameter mtry=1,...,max_mtry dynamiskt
  if (method=='rf'){
    num=num_covars
    max_mtry=ceiling(num*prop_mtry)
    params=expand.grid(seq(1,max_mtry))
    names(params)='mtry'
  }
  # --
  model=grid_search(method,params,inner_train_folds,inner_val_folds,fold2,
                    fold_recipe,i,thresholds)
  # Bästa kappa från grid search
  perf=model$kappa
  # Spara resultatet
  best_performance=perf
  best_covars=covars # alla kovariater
  best_model=model # modell, sannolikhetströskelvärde och kappa
  # ----- Backward stepwise selection ---------------------------
  for(p in 1:(num_covars-1)){
    performance=-1 # reset, bästa kappa för viss modellstorlek
    best_covar_subset=c() # reset, bästa nya reducerade modell
    # -- Om method=rf, bestäm parameter mtry=1,...,max_mtry dynamiskt
    if (method=='rf'){
      num=num_covars-p
      max_mtry=ceiling(num*prop_mtry) # ceiling, annars funkar ej om endast 1 kovariat
      params=expand.grid(seq(1,max_mtry))
      names(params)='mtry'
    }
    # --
    for(c in covar_names){
      del_idx=which(covar_names==c) # index för kovariaten som temporärt excluderas
      covar_sel=covar_names[-del_idx] # modell med kovariat c exluderad
      df_recipe=create_recipe(fold,covar_sel) # covar_sel
      fold2=df_recipe$df # data med utvalda kovariater
      fold_recipe=df_recipe$recipe # för standardisering av data baserat på träningsfold
      model=grid_search(method,params,inner_train_folds,inner_val_folds,fold2,
                        fold_recipe,i,thresholds)
      # Bästa kappa från grid search
      perf=model$kappa
      # Hitta och spara bästa reducerade modell
      if (perf > performance){
        performance=perf
        best_covar_subset=covar_sel # reducerade modellen
        mod=model # spara modellen som tränats på hela fold2
      }
    }
    # Kontrollera om bättre än eller lika bra som större modellen
    if (performance >= best_performance){
      best_performance=performance
      best_covars=best_covar_subset # nya bästa modellen
      best_model=mod
    } else {
      break
    }
    covar_names=best_covar_subset # uppdatera listan på möjliga kovariater som kan exkluderas
```

```
535
536     }
537     return(list(best_model=best_model,best_covars=best_covars,params=params))
538   } else {
539     # --- FORWARD STEPWISE SELECTION ---
540     for(p in 1:num_covars){
541       performance=-1 # reset, bästa kappa för viss modellstorlek
542       covar_name=c() # reset, bästa nya kovariat att lägga till
543       # -- Om method=rf, bestäm parameter mtry=1,...,max_mtry dynamiskt
544       if (method=='rf'){
545         num=p
546         max_mtry=ceiling(num*prop_mtry)
547         params=expand.grid(seq(1,max_mtry))
548         names(params)='mtry'
549       }
550       # --
551       for(c in covar_names){
552         covar_sel=c(best_covars,c)
553         df_recipe=create_recipe(fold,covar_sel) # covar_sel
554         fold2=df_recipe$df # data med utvalda kovariater
555         fold_recipe=df_recipe$recipe # för standardisering av data baserat på trä
    ningsfold
556         model=grid_search(method,params,inner_train_folds,inner_val_folds,fold2,
557                           fold_recipe,i,thresholds)
558         # Bästa kappa från grid search
559         perf=model$kappa
560         # Hitta och spara bästa kovariat
561         if (perf > performance){
562           performance=perf
563           covar_name=c # kovariaten som valdes
564           mod=model # spara modellen som tränats på hela fold2
565         }
566       }
567       # Kontrollera om bättre än p-1 modellen
568       if (performance > best_performance){
569         best_performance=performance
570         best_covars=c(best_covars,covar_name) # nya bästa modellen
571         best_model=mod
572       } else {
573         break
574       }
575
576       del_idx=which(covar_names==covar_name) # index för kovariaten som valdes
577       covar_names=covar_names[-del_idx] # uppdatera listan på möjliga kovariater
    att välja
578     }
579     return(list(best_model=best_model,best_covars=best_covars,params=params))
580   }
581 }
582
583 ######################################################################
584
585 ### B-08 class_converter ###########################################
586
587 class_converter=function(prob,threshold){
588   #-------------------------------------------------------------
589   # Omvandlar sannolikheter till klassificeringar
590   #-------------------------------------------------------------
591   if (prob >= threshold){
592     return('MNA')
593   } else {
594     return('Controls')
595   }
596 }
597
598 ######################################################################
599
600 ### B-09 outer_cv ##################################################
601
602 outer_cv=function(method,params,covars,data,prop_unknown,backward,num_outer_folds,
```

```
  num_inner_folds,seed,thresholds){
#-----------------------------------------------------------------
# yttre korsvalidering
# method: klassificeringsmetoden, t.ex. 'knn'
# params: grid av hyperparametrar, beror på vald klassificeringsmetod
# (se: https://topepo.github.io/caret/available-models.html)
# covars: vektor med kovariaternas namn
# data: vilken typ av mätningar, foot eller calf
# prop_unknown: andel (tal mellan 0 och 1) av obekräftade som tas med i trä
  ningsseten
# backward: TRUE om backward stepwise selection, FALSE om forward stepwise
  selection
# num_outer_folds: tal, antal yttre folds
# num_inner_folds: tal, antal inre folds
# seed: tal, sätter seed vilket ger samma slumpmässighet varje körning
# thresholds: vektor med möjliga sannolikhetströskelvärden
# (högre sannolikheter klassificeras som 'MNA')
# RETURNERAR:
# --- Alla argument från funktionsanropet
# $method
# $params
# $covars
# $data
# $prop_unknown
# $backward
# $num_outer_folds
# $num_inner_folds
# $seed
# $thresholds
# ---
# $sample_idx: index för de obekräftat sjuka som använts till träning
# (OBS 1: returnerar inget om obekräftat sjuka ej inkluderats i träning)
# (OBS 2: index syftar på index för observation från foot eller calf, ej df)
# $sel_covariates: lista med kovariater som valdes per modell
# $sel_hyperparams: lista med hyperparametrar som valdes per modell
# $sel_thresholds: vektor med sannolikhetströskelvärden som valdes per modell
# $kappas: vektor med kappa per modell baserat på yttre testmängd
# $confusion_matrix: förvirringsmatrix baserat på alla klassificeringarna
# i yttre korsvalideringen
#-----------------------------------------------------------------

# --- Lägg ev. till obekräftade sjuka ---
idx_unknown=which(data$label=='unknown') # index för obekräftade
set.seed(seed) # få samma "slumpmässighet" i varje körning
sample_idx=sample(idx_unknown,floor(length(idx_unknown)*prop_unknown)) #
  stickprov av index
# lägg ev. till stickprovet
data=rbind(data[data$label=='control' | data$label=='neuropathic',],data[
  sample_idx,])
row.names(data)=seq(1,dim(data)[1]) # uppdatera radindex

print(paste('Andel obekräftade:',prop_unknown,sep=" "))

# --- Nästlad korsvalidering ---

# --Dela upp i m yttre test folds - stratifierat
seed_init=seed
set.seed(seed)
fold_data = fold(data,k=num_outer_folds,cat_col='label',id_col='id') # OBS: inkl
  unknown
# skapa tom lista för vektorer med radindex per fold
outer_folds = vector("list",num_outer_folds)
for(i in 1:num_outer_folds){
  outer_folds[[i]]=which(fold_data$.folds == i) # vektor med radindex för fold i
}
# --

# Skapa tomma vektorer/listor för att spara bästa inställningar från varje inre
  korsvalidering samt kappa
sel_covariates=vector("list",num_outer_folds)
```

```
  sel_hyperparams=vector("list",num_outer_folds)
  sel_thresholds=rep(0,num_outer_folds)
  kappas=rep(0,num_outer_folds)
  predicted_classes=c()
  true_classes=c()
  mod_num=1
  for (outer_fold_idx in outer_folds){
    fold=data[-outer_fold_idx,] # data i yttre träningsfold
    row.names(fold)=seq(1,dim(fold)[1]) # uppdatera radindex
    test_fold=data[outer_fold_idx,] # data i yttre testfold, OBS: inkl. unknown
    test_fold=test_fold[test_fold$label!='unknown',] # exkludera unknown
    # Inre korsvalidering
    model=inner_cv(method,params,covars,fold,seed,num_inner_folds,backward,
    thresholds)
    # RESULTAT
    params=model$params
    # mer info om modellen som selekterades
    print(paste(method,'modell',mod_num,sep=" "))
    print(model$best_covars) # vilka kovariater som selekterades
    sel_covariates[[mod_num]]=model$best_covars
    # Hyperparametrarna som valdes
    for(n in 1:dim(params)[2]){
      print(names(params)[n])
      print(model$best_model$mod$results[,n])
    }
    sel_hyperparams[[mod_num]]=model$best_model$mod$results[,1:dim(params)[2]]
    # Sannolikhetströskelvärde
    print('Sannolikhetströskelvärde:')
    print(model$best_model$threshold)
    sel_thresholds[mod_num]=model$best_model$threshold
    # Sannolikheter och klassificering
    pred_probs=predict(model$best_model$mod,newdata=test_fold,type='prob') # testa
    modellen i yttre korsvalideringen
    pred_class=sapply(pred_probs$MNA,class_converter,threshold=
    model$best_model$threshold)
    true_class=test_fold$condition # OBS: condition EJ label
    # Kappa
    print('Kappa:')
    kappa_val=as.numeric(defaultSummary(data.frame(obs=true_class,pred=pred_class))
    [2])
    kappas[mod_num]=kappa_val # spara värdet
    print(kappa_val)
    # Spara pred_class & true_class - för att skapa EN förvirringsmatris
    predicted_classes=c(predicted_classes,as.character(pred_class))
    true_classes=c(true_classes,as.character(true_class))
    mod_num=mod_num+1
    seed=seed+1
  }
  # Förvirringsmatris
  confusion_matrix=confusionMatrix(data=factor(predicted_classes,levels=c('Controls
    ','MNA')),
                                   reference=factor(true_classes,levels=c('Controls
    ','MNA')),
                                   positive='MNA')
  print(confusion_matrix) # OBS: MNA ej neuropathic

  # Returnera
  return(list(method=method,
              params=params,
              covars=covars,
              data=data,
              prop_unknown=prop_unknown,
              backward=backward,
              num_outer_folds=num_outer_folds,
              num_inner_folds=num_inner_folds,
              seed=seed_init,
              thresholds=thresholds,
              sample_idx=sample_idx,
              sel_covariates=sel_covariates,
              sel_hyperparams=sel_hyperparams,
```

```
                sel_thresholds=sel_thresholds,
                kappas=kappas,
                confusion_matrix=confusion_matrix))
}

#####################################################################

### C-01 knn ########################################################

###### ------------------------------------------------------------------------
# Method & params
method='knn'
params=expand.grid(k=c(1,3,5))
# Kovariater: ändra om det behövs
covars=covariates
# Vilken typ av mätningar
data=foot
# Andel av obekräftade som tas med i träningsseten
prop_unknown=0
# Backward stepwise selection?
backward=FALSE
### Antal yttre folds
num_outer_folds=3
### Antal inre folds
num_inner_folds=3
### Sätt seed, ger samma slumpmässighet varje körning
seed=2
### Möjliga sannolikhetströskelvärden - högre sannolikheter klassificeras som 'MNA'
thresholds=c(0.2,0.3,0.4)
###### ------------------------------------------------------------------------
res=outer_cv(method,params,covars,data,prop_unknown,backward,num_outer_folds,
    num_inner_folds,seed,thresholds)

res$method
res$params
res$covars
head(res$data)
res$prop_unknown
res$backward
res$num_outer_folds
res$num_inner_folds
res$seed
res$thresholds
res$sample_idx # returnerar inget eftersom prop_unknown=0
res$sel_covariates
res$sel_hyperparams
res$sel_thresholds
res$kappas
res$confusion_matrix

#####################################################################

### C-02 rf  ########################################################

###
method='rf'
params=NULL
data=foot
###

covars=covariates
prop_unknown=0
backward=FALSE
num_outer_folds=3
num_inner_folds=3
seed=2
thresholds=c(0.1,0.2,0.3,0.4,0.5,0.6)

res=outer_cv(method,params,covars,data,prop_unknown,backward,num_outer_folds,
    num_inner_folds,seed,thresholds)
```

```
798
799 ######################################################################
800
801 ### D-01 Bästa metod på fot ##################################
802
803 sample_idx=res$sample_idx # index för obekräftat sjuka som använts till träning
804
805 ### -- Träning- & testmängder ------------------
806 data=foot
807 if (length(sample_idx) > 0){
808   data=data[-sample_idx,] # exkludera obekräftade som tidigare använts till träning
809 }
810 train=data[data$label!='unknown',]
811 row.names(train)=seq(1,dim(train)[1]) # uppdatera radindex
812 test=data[data$label=='unknown',]
813 ### ----------------------------------------------
814
815 ### --- Selektera modell med korsvalidering ---
816
817 # Ange metod, parametrar, forward/backward ##########
818 method='rf' # 'rf' eller 'knn'
819 params=NULL
820 backward=FALSE # TRUE=stegvis bakåtselektion, FALSE=stegvis framåtselektion
821 ######################################################################
822
823 covars=covariates
824 fold=train # träningsdata
825 i=1 # seed
826 num_inner_folds=3 # OBS: 3 för fot
827 thresholds=c(0.1,0.2,0.3,0.4,0.5,0.6)
828
829 mod=inner_cv(method,params,covars,fold,i,num_inner_folds,backward,thresholds)
830 ### ----------------------------------------------
831
832 ### --- Modellen som selekterades ----------------
833 print('Kovariater:')
834 print(mod$best_covars)
835 print('Sannolikhetströskelvärde:')
836 print(mod$best_model$threshold)
837
838 parameters=mod$params
839 for(n in 1:dim(parameters)[2]){
840   print(names(parameters)[n])
841   print(mod$best_model$mod$results[,n])
842 }
843 ### ----------------------------------------------
844
845 ### --- Testa modellen på obekräftat sjuka -----
846 pred_probs=predict(mod$best_model$mod,newdata=test,type='prob') # sannolikheter att
      MNA
847 pred_class_foot=sapply(pred_probs$MNA,class_converter,threshold=
     mod$best_model$threshold) # klassificering
848 foot_test=test # copy
849 foot_test$class=pred_class_foot # klasificering av obekräftat sjuka
850 id_unknown_foot=foot_test$id # id för obekräftat sjuka
851
852 ######################################################################
853
854 ### D-02 Bästa metod på vad ##################################
855
856 sample_idx=res$sample_idx # index för obekräftat sjuka som använts till träning
857
858 ### -- Träning- & testmängder ------------------
859 data=calf
860 if (length(sample_idx) > 0){
861   data=data[-sample_idx,] # exkludera obekräftade som tidigare använts till träning
862 }
863 train=data[data$label!='unknown',]
864 row.names(train)=seq(1,dim(train)[1]) # uppdatera radindex
865 test=data[data$label=='unknown',]
```

```
### ------------------------------------------------

### --- Selektera modell med korsvalidering ---

# Ange metod, parametrar, forward/backward #########
method='rf' # 'rf' eller 'knn'
params=NULL
backward=FALSE # TRUE=stegvis bakåtselektion, FALSE=stegvis framåtselektion
################################################################

covars=covariates
fold=train # träningsdata
i=1 # seed
num_inner_folds=5 # OBS: 5 för vad
thresholds=c(0.1,0.2,0.3,0.4,0.5,0.6)

mod=inner_cv(method,params,covars,fold,i,num_inner_folds,backward,thresholds)
### ------------------------------------------------

### --- Modellen som selekterades ----------------
print('Kovariater:')
print(mod$best_covars)
print('Sannolikhetströskelvärde:')
print(mod$best_model$threshold)

parameters=mod$params
for(n in 1:dim(parameters)[2]){
  print(names(parameters)[n])
  print(mod$best_model$mod$results[,n])
}
### ------------------------------------------------

### --- Testa modellen på obekräftat sjuka -----
pred_probs=predict(mod$best_model$mod,newdata=test,type='prob') # sannolikheter att MNA
pred_class_calf=sapply(pred_probs$MNA,class_converter,threshold=mod$best_model$threshold) # klassificering
calf_test=test # copy
calf_test$class=pred_class_calf # klasificering av obekräftat sjuka
id_unknown_calf=calf_test$id # id för obekräftat sjuka

################################################################

### D-03 Jämförelse av modeller för fot och vad ############

### JÄMFÖRELSE AV MODELLER FÖR FOT OCH VAD ###
# genom mätningar gjorda på samma personer
same_id=intersect(id_unknown_foot,id_unknown_calf) # gemensamma id för mätningar på fot och vad

print('Klassificering av vadmätningar:')
print(calf_test[match(same_id, calf_test$id),]$class)
print('Klassificering av fotmätningar:')
print(foot_test[match(same_id, foot_test$id),]$class)

confusion_matrix=confusionMatrix(data=factor(foot_test[match(same_id, foot_test$id),]$class,levels=c('Controls','MNA')),
                                 reference=factor(calf_test[match(same_id, calf_test$id),]$class,levels=c('Controls','MNA')),
                                 positive='MNA')
print(confusion_matrix)

################################################################

### D-04 Bästa metod på all data ###########################

### BÄSTA METOD PÅ ALL DATA ###

sample_idx=res$sample_idx # index för obekräftat sjuka som använts till träning

```

```
### -- Träning- & testmängder ----------------
data=df
if (length(sample_idx) > 0){
  data=data[-sample_idx,] # exkludera obekräftade som tidigare använts till träning
}
train=data[data$label!='unknown',]
row.names(train)=seq(1,dim(train)[1]) # uppdatera radindex
test=data[data$label=='unknown',]
### ---------------------------------------

### --- Selektera modell med korsvalidering ---

# Ange metod, parametrar, forward/backward ###########
method='rf' # 'rf' eller 'knn'
params=NULL
backward=FALSE # TRUE=stegvis bakåtselektion, FALSE=stegvis framåtselektion
#################################################################

covars=covariates
fold=train # träningsdata
i=1 # seed
num_inner_folds=5 # OBS: 5 för all data
thresholds=c(0.1,0.2,0.3,0.4,0.5,0.6)

mod=inner_cv(method,params,covars,fold,i,num_inner_folds,backward,thresholds)
### ---------------------------------------

### --- Modellen som selekterades -------------
print('Kovariater:')
print(mod$best_covars)
print('Sannolikhetströskelvärde:')
print(mod$best_model$threshold)

parameters=mod$params
for(n in 1:dim(parameters)[2]){
  print(names(parameters)[n])
  print(mod$best_model$mod$results[,n])
}
### ---------------------------------------

### --- Testa modellen på obekräftat sjuka ----
pred_probs=predict(mod$best_model$mod,newdata=test,type='prob') # sannolikheter att
    MNA
pred_class_all=sapply(pred_probs$MNA,class_converter,threshold=
    mod$best_model$threshold) # klassificering
print(pred_class_all)

#################################################################
#################################END#############################
```

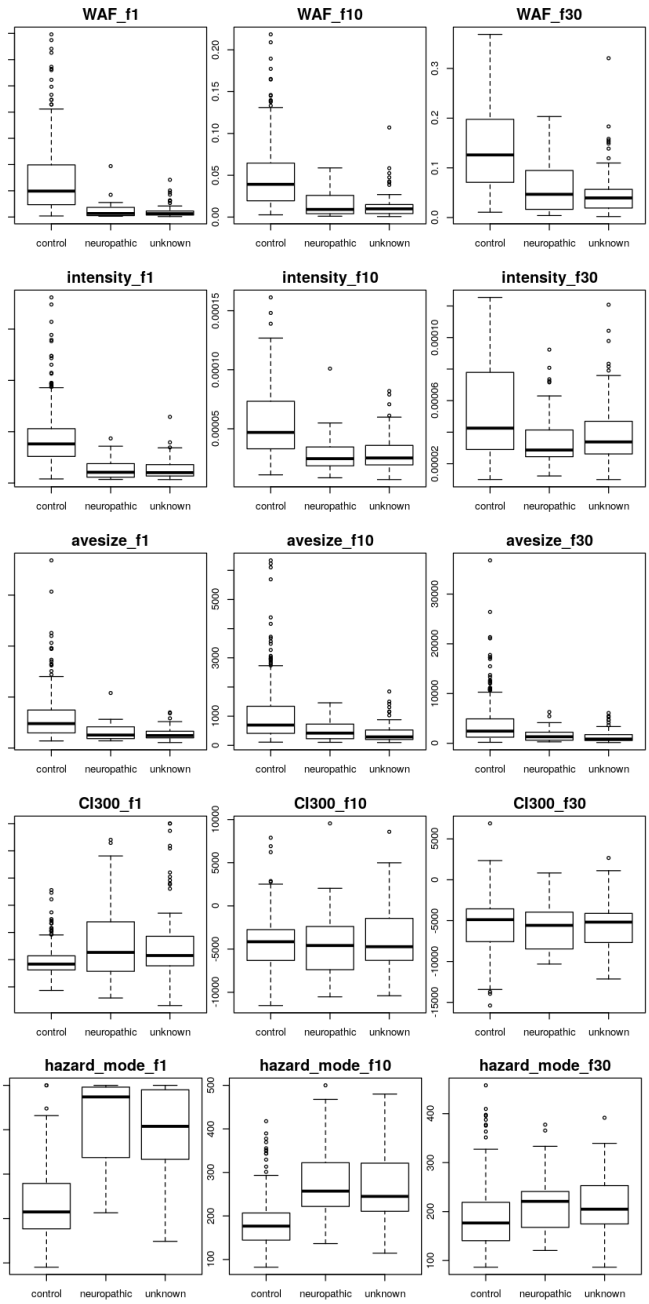
Bilaga 5 : Algoritm för utvärdering av en klassificeringsmetod på ett visst mätområde med nästlad korsvalidering

Nedan följer en schematisk beskrivning av den algoritm för utvärdering av en klassificeringsmetod på ett specifikt mätområde som vi använder i detta projekt. Denna algoritm är ett komplement till beskrivningen i avsnitt 4.2 *Utvärdering och jämförelse av klassificeringsmetoder*.

1. Definiera datamängd D som observationer av bekräftat sjuka och friska från ett visst mätområde
2. Lägg eventuellt till ett stickprov av de obekräftat sjuka från samma mätområde till datamängd D
3. Dela upp datamängd D i m stycken indelningar som bildar m yttre träningsmängder med motsvarande m yttre testmängder
   (a) Om obekräftat sjuka inkluderats, exkluderas dessa observationer från de m yttre testmängderna
4. För varje yttre träningsmängd
   (a) Dela upp den yttre träningsmängden i n indelningar som bildar n inre träningsmängder med motsvarande n inre valideringsmängder
      i. Om obekräftat sjuka inkluderats, exkluderas dessa observationer från de n inre valideringsmängderna
   (b) Använd stegvis framåt- eller bakåtselektion för att välja vilka kovariater som ska ingå i modellen genom att på varje mängd av kovariater som undersöks utföra inre korsvalidering
      i. Skapa otränade modeller utifrån alla kombinationer av valbara parametrar
      ii. Träna varje modell på de n inre träningsmängderna och validera med de motsvarande n inre valideringsmängdera och spara sannolikheten att vara sjuk som modellen anger per observation
      iii. Tillämpa de olika sannolikhetströskelvärdena på sannolikheterna för att generera klassificeringar
      iv. Välj den modell som ger högsta minsta kappa på en valideringsmängd
   (c) Notera vilka kovariater, valbara parametrar och sannolikhetströskelvärden som ingår i den bästa modellen, d.v.s. den med högsta minsta kappa, från stegvis framåt- eller bakåtselektion
   (d) Träna modellen med dessa kovariater och valbara parametrar på hela yttre träningsmängden
5. Testa de m modellerna (som tränats på de m yttre träningsmänderna) på de m motsvarande yttre testmängderna och spara sannolikheten att vara sjuk som modellen anger per observation
6. Tillämpa sannolikhetströskelvärdet (som identifieras som optimalt i de m yttre träningsmängderna) på sannolikheterna från de motsvarande m yttre testmängderna för att generera klassificeringar
7. Beräkna kappa för de m yttre testmängderna samt totalt kappa
8. Returnera
   (a) Vilka kovariater, valbara parametrar och sannolikhetströskelvärde som ingick i de m valda modellerna samt kappavärdet från de yttre testmängderna
   (b) En förvirringsmatris baserad på en sammanslagning av klassificeringarna av de m yttre testmändgerna samt totalt kappa
   (c) Om ett stickprov av obekräftat sjuka används vid träning, vilka observationer som ingick i stickprovet

Bilaga 6 : Beskrivande analys av kovariaterna

Gruppen med obekräftat sjuka innehåller möjligtvis en blandning av sjuka och friska försökspersoner. Därför undersöktes mätvärdena för de 15 kovariaterna (baserat på observationer från hela datamängden) per grupp av försökspersoner med hjälp av låddiagram (fig. 13). Grupperna med bekräftat sjuka (neuropathic) och obekräftat sjuka (unknown) uppvisar generellt liknande mätvärden (se figur 13). För vissa kovariater, såsom **WAF_f1** och **Intensity_f1**, kan dessa två grupper enklare särskiljas visuellt från gruppen med friska försökspersoner (control), medan alla tre grupperna uppvisar liknande mätvärden för andra kovariater, t.ex. **CI300_f30**. Alla kovariater kan dock vara av värde eftersom kombinationer av kovariater möjligtvis kan ge god separation mellan grupperna i högre dimensioner.



Figur 13: Låddiagram per kovariat för friska försökspersoner (control), bekräftat sjuka (neuropathic) och obekräftat sjuka (unknown) baserat på hela datamängden. Bekräftat och obekräftat sjuka uppvisar generellt liknande mätvärden.

Bilaga 7 : Resultat för körningar med klassificeringsmetoden: k-NN

Nedan redovisas resultatet för olika körningar med k-NN algoritmen. I tabellerna 8, 9 och 10 är 'k' parametern i k-NN, d.v.s. de 'k' som testas i korsvalideringen. Två olika sökmetoder används, stegvis framåtselektion och stegvis bakåtselektion. Antalet indelningar i korsvalideringen beror på vilken data som används. För vadmätningar- och vad & fotmätningar så är antalet indelningar 5, och för enbart fotmätningar så är antalet indelningar 3.

Fot- & vadmätningar

I tabell 8 ser vi resultatet för de olika körningarna med k-NN där all data har använts. Stegvis bakåtselektion genererar ett betydligt högre totalt kappa gentemot stegvis framåtselektion. Vi ser ett högsta totalt kappa: 0.5823, detta bekräftas till viss del med körningarna med annan `seed`.

Tabell 8: Resultat från klassificering med k-NN fot- &vadmätningar.

Körning	**k**	**Andel obekräftade**	**Framåt-/ bakåt-selektion**	**indel-ningar**	**seed**	**Totalt kappa** (högsta minsta kappa användes i inre korsvalideringen)
1	1,3,5,7,9,11, 13,15,17,19	0	Stegvis-framåtselektion	5	2	**0.1711**
1	1,3,5,7,9,11, 13,15,17,19	0	Stegvis-bakåtselektion	5	2	**0.5823**
2	1,3,5,7,9,11, 13,15,17,19	0	Stegvis-framåtselektion	5	3	**0.1399**
2	1,3,5,7,9,11, 13,15,17,19	0	Stegvis-bakåtselektion	5	3	**0.4502**
3	1,3,5,7,9,11, 13,15,17,19	0	Stegvis-framåtselektion	5	1	**0.1211**
3	1,3,5,7,9,11, 13,15,17,19	0	Stegvis-bakåtselektion	5	1	**0.4458**
4	1,3,5,7,9,11, 13,15,17,19	0.3	Stegvis-framåtselektion	5	2	**0.1711**
4	1,3,5,7,9,11, 13,15,17,19	0.3	Stegvis-bakåtselektion	5	2	**0.3735**
5	1,3,5,7,9,11, 13,15,17,19	0.3	Stegvis-framåtselektion	5	3	**0.0795**
5	1,3,5,7,9,11, 13,15,17,19	0.3	Stegvis-bakåtselektion	5	3	**0.3498**
6	1,3,5,7,9,11, 13,15,17,19	0.3	Stegvis-framåtselektion	5	1	**-0.0509**
6	1,3,5,7,9,11, 13,15,17,19	0.3	Stegvis-bakåtselektion	5	1	**0.4045**

		Referens	
		Frisk	Sjuk
Prediktion	Frisk	292	11
	Sjuk	9	16

Figur 14: Förvirringsmatris för körning 1 med stegvis bakåtselektion från tabell 8.

Vadmätningar

I tabell 9 ser vi resultatet från de olika körningarna med k-NN där data från vad har använts. Vilken sökmetod som ger bäst totalt kappa kan inte ses från de olika körningarna.

Tabell 9: Resultat från klassificering med k-NN där data från **vad** har använts.

Körning	**k**	**Andel obekräftade**	**Framåt-/ bakåt-selektion**	**indelningar**	**seed**	**Totalt kappa** (högsta minsta kappa användes i inre korsvalideringen)
1	1,3,5,7, 9,11,13	0	Stegvis-framåtselektion	5	2	**0.2649**
1	1,3,5,7, 9,11,13	0	Stegvis-bakåtselektion	5	2	**0.1980**
2	1,3,5,7, 9,11,13	0	Stegvis-framåtselektion	5	3	**0.3639**
2	1,3,5,7, 9,11,13	0	Stegvis-bakåtselektion	5	3	**0.3504**
3	1,3,5,7, 9,11,13	0	Stegvis-framåtselektion	5	1	**0.2720**
3	1,3,5,7, 9,11,13	0	Stegvis-bakåtselektion	5	1	**0.4242**
4	1,3,5,7, 9,11,13	0.3	Stegvis-framåtselektion	5	2	**-0.0118**
4	1,3,5,7, 9,11,13	0.3	Stegvis-bakåtselektion	5	2	**0.6275**
5	1,3,5,7, 9,11,13	0.3	Stegvis-framåtselektion	5	3	**0.5033**
5	1,3,5,7, 9,11,13	0.3	Stegvis-bakåtselektion	5	3	**0.4809**
6	1,3,5,7, 9,11,13	0.3	Stegvis-framåtselektion	5	1	**0.3277**
6	1,3,5,7, 9,11,13	0.3	Stegvis-bakåtselektion	5	1	**0.4352**

		Referens	
		Frisk	Sjuk
Prediktion	Frisk	135	6
	Sjuk	18	12

Figur 15: Förvirringsmatris för körning 3 med stegvis bakåtselektion från tabell 9.

Fotmätningar

I tabell 10 ser vi resultatet för de olika körningarna med k-NN där data från fot har använts. Stegvis bakåtselektion genererar ett betydligt högre totalt kappa gentemot stegvis framåtselektion...

Tabell 10: Resultat från klassificering med k-NN där data från **fot** har använts.

Körning	**k**	**Andel obekräftade**	**Framåt-/ bakåt-selektion**	**indelningar**	**seed**	**Totalt kappa** (högsta minsta kappa användes i inre korsvalideringen)
1	1,3,5,7	0	Stegvis-framåtselektion	3	2	**0.1337**
1	1,3,5,7	0	Stegvis-bakåtselektion	3	2	**0.4614**
2	1,3,5,7	0	Stegvis-framåtselektion	3	3	**0.1337**
2	1,3,5,7	0	Stegvis-bakåtselektion	3	3	**0.4614**
3	1,3,5,7	0	Stegvis-framåtselektion	3	1	**-0.0166**
3	1,3,5,7	0	Stegvis-bakåtselektion	3	1	**0.1063**
4	1,3,5,7	0.3	Stegvis-framåtselektion	3	2	**0.0917**
4	1,3,5,7	0.3	Stegvis-bakåtselektion	3	2	**0.1224**
5	1,3,5,7	0.3	Stegvis-framåtselektion	3	3	**0.4179**
5	1,3,5,7	0.3	Stegvis-bakåtselektion	3	3	**0.1462**
6	1,3,5,7	0.3	Stegvis-framåtselektion	3	1	**0.2824**
6	1,3,5,7	0.3	Stegvis-bakåtselektion	3	1	**0.1749**

		Referens	
		Frisk	Sjuk
Prediktion	Frisk	139	3
	Sjuk	9	6

Figur 16: Förvirringsmatris för körning 2 med stegvis bakåtselektion från tabell 10.

Bilaga 8 : Resultat för körningar med klassificeringsmetoden: Slumpmässig skog

Nedan redovisas resultatet för olika körningar med algoritmen slumpmässig skog. Parametern för denna algoritm är *mtry*, vilket är en parameter som ändras dynamiskt. En annan inställning som är förutbestämd är hur många kovariater som finns tillgängliga att välja bland i varje beslutnod. Denna är manuellt satt till 0.4, det vill säga 40 % av kovariaterna väljs slumpvis vid varje beslutsnod.

Fot- & vadmätningar

I tabell 11 redovisas resultatet för klassificering med slumpmässig skog på fot- & vadmätningar. Vi ser ett något högre kappa när sökmetoden stegvis bakåtselektion används. Högst totalt kappa (0.6100) erhålls då andelen obekräftade är 0.3 och sökmetoden stegvis framåtselektion används.

Tabell 11: Resultat från klassificering med slumpmässig skog där data från **fot & vad** har använts.

Körning	**Andel obekräftade**	**Framåt-/ bakåt-selektion**	**indelningar**	**seed**	**Totalt kappa** (högsta minsta kappa användes i inre korsvalideringen)
1	0	Stegvis-framåtselektion	5	2	**0.5372**
1	0	Stegvis-bakåtselektion	5	2	**0.4869**
2	0	Stegvis-framåtselektion	5	3	**0.4988**
2	0	Stegvis-bakåtselektion	5	3	**0.5653**
3	0	Stegvis-framåtselektion	5	1	**0.5111**
3	0	Stegvis-bakåtselektion	5	1	**0.4766**
4	0.3	Stegvis-framåtselektion	5	2	**0.6100**
4	0.3	Stegvis-bakåtselektion	5	2	**0.4919**
5	0.3	Stegvis-framåtselektion	5	3	**0.4603**
5	0.3	Stegvis-bakåtselektion	5	3	**0.4723**
6	0.3	Stegvis-framåtselektion	5	1	**0.5085**
6	0.3	Stegvis-bakåtselektion	5	1	**0.5198**

		Referens	
		Frisk	Sjuk
Prediktion	Frisk	292	10
	Sjuk	9	17

Figur 17: Förvirringsmatris för körning 4 med stegvis framåtselektion från tabell 11.

Vadmätningar

Resultatet för klassificering med slumpmässig skog för vadmätningar ses i tabell 12. Bäst resultat erhålls för körning nummer 3, med stegvis bakåtselektion. Om vi inkluderar 30 % av andelen obekräftade ses ingen förbättring.

Tabell 12: Resultat från klassificering med slumpmässig skog där data från **vad** har använts.

Körning	**Andel obekräftade**	**Framåt-/ bakåt-selektion**	**indelningar**	**seed**	**Totalt kappa** (högsta minsta kappa användes i inre korsvalideringen)
1	0	Stegvis-framåtselektion	5	2	**0.3804**
1	0	Stegvis-bakåtselektion	5	2	**0.5863**
2	0	Stegvis-framåtselektion	5	3	**0.6082**
2	0	Stegvis-bakåtselektion	5	3	**0.5632**
3	0	Stegvis-framåtselektion	5	1	**0.4848**
3	0	Stegvis-bakåtselektion	5	1	**0.6275**
4	0.3	Stegvis-framåtselektion	5	2	**0.4412**
4	0.3	Stegvis-bakåtselektion	5	2	**0.5265**
5	0.3	Stegvis-framåtselektion	5	3	**0.4150**
5	0.3	Stegvis-bakåtselektion	5	3	**0.5265**
6	0.3	Stegvis-framåtselektion	5	1	**0.5365**
6	0.3	Stegvis-bakåtselektion	5	1	**0.5084**

		Referens	
		Frisk	Sjuk
Prediktion	Frisk	147	6
	Sjuk	6	12

Figur 18: Förvirringsmatris för körning 3 med stegvis bakåtselektion från tabell 12.

Fotmätningar

Resultatet för klassificering med slumpmässig skog för fotmätningar ses i tabell 13. Bäst resultat erhålls för körning nummer 1, med stegvis bakåtselektion. Om vi inkluderar 30 % av andelen obekräftade ses en marginell förbättring

Tabell 13: Resultat från klassificering med slumpmässig skog där data från **fot** har använts.

Körning	**Andel obekräftade**	**Framåt-/ bakåt-selektion**	**indelningar**	**seed**	**Totalt kappa** (högsta minsta kappa användes i inre korsvalideringen)
1	0	Stegvis-framåtselektion	5	2	**0.1337**
1	0	Stegvis-bakåtselektion	5	2	**0.4663**
2	0	Stegvis-framåtselektion	5	3	**0.3985**
2	0	Stegvis-bakåtselektion	5	3	**0.3183**
3	0	Stegvis-framåtselektion	5	1	**0.3716**
3	0	Stegvis-bakåtselektion	5	1	**0.2314**
4	0.3	Stegvis-framåtselektion	5	2	**0.3376**
4	0.3	Stegvis-bakåtselektion	5	2	**0.3525**
5	0.3	Stegvis-framåtselektion	5	3	**0.2508**
5	0.3	Stegvis-bakåtselektion	5	3	**0.4959**
6	0.3	Stegvis-framåtselektion	5	1	**0.0788**
6	0.3	Stegvis-bakåtselektion	5	1	**0.4404**

		Referens	
		Frisk	Sjuk
Prediktion	Frisk	142	4
	Sjuk	6	5

Figur 19: Förvirringsmatris för körning 1 med stegvis bakåtselektion från tabell 13.

Bilaga 9 : Resultat för körningar med klassificeringsmetoden: Neurala nätverk

Nedan redovisas resultatet för klassificering med neurala nätverk. Metoden som används i R heter 'nnet' och har två parametrar att ställa in, antalet noder och viktnedbrytning (weight decay). Dessa kan i vårat fall anta tre värden, # noder = (2,5,10) och viktnedbrytning = (1e-1,1e-4,0).

Fot- & vadmätningar

Resultatet för klassificering med neurala nätverk för fot- och vadmätningar ses i tabell 14. Bäst resultat erhålls för körning nummer 1, med stegvis framåtselektion. Någon tendens till ett jämnare

värde på totalt kappa, d.v.s. ingen stor variation mellan de olika körningarna. Ingen förbättring kan ses då andelen obekräftade inkluderas.

Tabell 14: Resultat från klassificeringen med neurala nätverk där data från **fot & vad** har använts.

Körning	**Weight decay**	**# Noder**	**Andel obekräftade**	**Framåt-/ bakåt-selektion**	**indel-ningar**	**seed**	**Totalt kappa** (högsta minsta kappa användes i inre korsvalideringen)
1	1e-1, 1e-4, 0	2,5,10	0	Stegvis-framåtselektion	5	2	**0.5959**
1	1e-1, 1e-4, 0	2,5,10	0	Stegvis-bakåtselektion	5	2	**0.4458**
2	1e-1, 1e-4, 0	2,5,10	0	Stegvis-framåtselektion	5	3	**0.5372**
2	1e-1, 1e-4, 0	2,5,10	0	Stegvis-bakåtselektion	5	3	**0.5127**
3	1e-1, 1e-4, 0	2,5,10	0	Stegvis-framåtselektion	5	1	**0.4350**
3	1e-1, 1e-4, 0	2,5,10	0	Stegvis-bakåtselektion	5	1	**0.4666**
4	1e-1, 1e-4, 0	2,5,10	0.3	Stegvis-framåtselektion	5	2	**0.4975**
4	1e-1, 1e-4, 0	2,5,10	0.3	Stegvis-bakåtselektion	5	2	**0.431**
5	1e-1, 1e-4, 0	2,5,10	0.3	Stegvis-framåtselektion	5	3	**0.4766**
5	1e-1, 1e-4, 0	2,5,10	0.3	Stegvis-bakåtselektion	5	3	**0.4869**
6	1e-1, 1e-4, 0	2,5,10	0.3	Stegvis-framåtselektion	5	1	**0.3733**
6	1e-1, 1e-4, 0	2,5,10	0.3	Stegvis-bakåtselektion	5	1	**0.5347**

		Referens	
		Frisk	Sjuk
Prediktion	Frisk	295	12
	Sjuk	6	15

Figur 20: Förvirringsmatris för körning 1 med stegvis framåtselektion från tabell 14.

Vadmätningar

Resultatet för klassificering med neurala nätverk för vadmätningar ses i tabell 15. Bäst resultat när andelen obekräftade är satt till 0 erhålls för körning nummer 1, detta med stegvis bakåtselektion. När andelen obekräftade inkluderas fås ett väldigt högt totalt kappa, 0.7085. Detta är det högsta totala kappa för alla körningar oavsett metod och inställningar. Lika högt totalt kappa kunde inte bekräftas i de andra körningarna med annan `seed` dock.

Tabell 15: Resultat från klassificeringen med neurala nätverk där data från **vad** har använts.

Körning	**Weight decay**	**# Noder**	**Andel obekräftade**	**Framåt-/ bakåt-selektion**	**indelningar**	**seed**	**Totalt kappa** (högsta minsta kappa användes i inre korsvalideringen)
1	1e-1, 1e-4, 0	2,5,10	0	Stegvis-framåtselektion	5	2	**0.3643**
1	1e-1, 1e-4, 0	2,5,10	0	Stegvis-bakåtselektion	5	2	**0.5670**
2	1e-1, 1e-4, 0	2,5,10	0	Stegvis-framåtselektion	5	3	**0.4750**
2	1e-1, 1e-4, 0	2,5,10	0	Stegvis-bakåtselektion	5	3	**0.4911**
3	1e-1, 1e-4, 0	2,5,10	0	Stegvis-framåtselektion	5	1	**0.4192**
3	1e-1, 1e-4, 0	2,5,10	0	Stegvis-bakåtselektion	5	1	**0.5654**
4	1e-1, 1e-4, 0	2,5,10	0.3	Stegvis-framåtselektion	5	2	**0.3030**
4	1e-1, 1e-4, 0	2,5,10	0.3	Stegvis-bakåtselektion	5	2	**0.7085**
5	1e-1, 1e-4, 0	2,5,10	0.3	Stegvis-framåtselektion	5	3	**0.4745**
5	1e-1, 1e-4, 0	2,5,10	0.3	Stegvis-bakåtselektion	5	3	**0.4865**
6	1e-1, 1e-4, 0	2,5,10	0.3	Stegvis-framåtselektion	5	1	**0.4865**
6	1e-1, 1e-4, 0	2,5,10	0.3	Stegvis-bakåtselektion	5	1	**0.5676**

		Referens	
		Frisk	Sjuk
Prediktion	Frisk	142	5
	Sjuk	11	13

Figur 21: Förvirringsmatris för körning 1 med stegvis bakåtselektion från tabell 15.

Fotmätningar

Resultatet för klassificering med neurala nätverk för fotmätningar ses i tabell 16. Bäst resultat erhålls för körning nummer 2, där båda sökmetoderna ger ett betydligt högre kappa gentemot de andra körningarna. Ingen förbättring kan ses då andelen obekräftade inkluderas.

Tabell 16: Resultat från klassificeringen med neurala nätverk där data från **fot** har använts.

Körning	**Weight decay**	**# Noder**	**Andel obekräftade**	**Framåt-/ bakåt-selektion**	**indelningar**	**seed**	**Totalt kappa** (högsta minsta kappa användes i inre korsvalideringen)
1	1e-1, 1e-4, 0	2,5,10	0	Stegvis-framåtselektion	3	2	**0.2529**
1	1e-1, 1e-4, 0	2,5,10	0	Stegvis-bakåtselektion	3	2	**0.1798**
2	1e-1, 1e-4, 0	2,5,10	0	Stegvis-framåtselektion	3	3	**0.5285**
2	1e-1, 1e-4, 0	2,5,10	0	Stegvis-bakåtselektion	3	3	**0.5414**
3	1e-1, 1e-4, 0	2,5,10	0	Stegvis-framåtselektion	3	1	**0.1421**
3	1e-1, 1e-4, 0	2,5,10	0	Stegvis-bakåtselektion	3	1	**0.1917**
4	1e-1, 1e-4, 0	2,5,10	0.3	Stegvis-framåtselektion	3	2	**0.2516**
4	1e-1, 1e-4, 0	2,5,10	0.3	Stegvis-bakåtselektion	3	2	**0.5011**
5	1e-1, 1e-4, 0	2,5,10	0.3	Stegvis-framåtselektion	3	3	**0.1749**
5	1e-1, 1e-4, 0	2,5,10	0.3	Stegvis-bakåtselektion	3	3	**0.5011**
6	1e-1, 1e-4, 0	2,5,10	0.3	Stegvis-framåtselektion	3	1	**0.4404**
6	1e-1, 1e-4, 0	2,5,10	0.3	Stegvis-bakåtselektion	3	1	**0.3712**

		Referens	
		Frisk	Sjuk
Prediktion	Frisk	142	3
	Sjuk	6	6

Figur 22: Förvirringsmatris för körning 2 med stegvis bakåtselektion från tabell 16.

Bilaga 10 : Klassificering av obekräftat sjuka

Nedan visas resultat för klassificering av obekräftat sjuka. Klassificeringen är gjord med den bästa modellen per mätområde. Dessa modeller ses i tabell 3 i resultatets huvuddel. Då dessa individers sjukdomstillstånd är obekräftat kan dessa resultat inte i vanlig mening bekräftas. Också värt att notera är att Id inte står för en specifik individs Id, utan är raden för utskrift i R. Klassificering som frisk står som 'Controls' i nedanstående tabeller, och klassificering som sjuk står som 'MNA'.

Tabell 17: Klassificering av obekräftat sjuka för bästa modell med data från **fot**.

Id 1-9	MNA	MNA	MNA	MNA	MNA	MNA	MNA	MNA	MNA
Id 10-18	MNA	MNA	MNA	MNA	MNA	Controls	MNA	Controls	MNA
Id 19-27	Controls	Controls	MNA	Controls	MNA	Controls	MNA	MNA	Controls
Id 28	Controls								

Tabell 18: Klassificering av obekräftat sjuka för bästa modell med data från **vad**.

Id 1-9	Controls	Controls	Controls	Controls	Controls	MNA	Controls	MNA	Controls
Id 10-18	Controls	Controls	Controls	Controls	MNA	Controls	MNA	Controls	Controls
Id 19-27	Controls	Controls	Controls	Controls	Controls	Controls	Controls	Controls	Controls
Id 28-36	Controls	Controls	Controls	Controls	Controls	MNA	Controls	Controls	Controls
Id 37-45	Controls	Controls	MNA	Controls	Controls	MNA	Controls	Controls	Controls

Tabell 19: Klassificering av obekräftat sjuka för bästa modell med data från **fot & vad**.

Id 1-9	MNA	MNA	Controls	MNA	Controls	MNA	Controls	Controls	MNA
Id 10-18	MNA	MNA	MNA	MNA	MNA	MNA	Controls	MNA	Controls
Id 19-27	Controls	Controls	MNA	Controls	MNA	MNA	Controls	Controls	MNA
Id 28-36	MNA	Controls	MNA	Controls	MNA	Controls	MNA	MNA	Controls
Id 37-45	Controls	Controls	Controls	MNA	Controls	Controls	MNA	Controls	Controls
Id 46-54	MNA	MNA	MNA	Controls	Controls	Controls	MNA	Controls	Controls
Id 55-63	Controls	MNA	Controls	MNA	MNA	MNA	Controls	MNA	MNA
Id 64-72	Controls	Controls	Controls	MNA	MNA	Controls	MNA	Controls	Controls
Id 73	Controls								

Jämförelse av modeller för fot och vad

Det är totalt 26 individer där data från både fot och vad finns tillgängligt. Nedan redovisas klassificeringen av dessa obekräftat sjuka individer.

Tabell 20: Klassificering obekräftat sjuka för bästa modell med data från **fot** för de 26 individer där data från båda mätområden finns tillgängligt.

Id 1-9	MNA	MNA	MNA	MNA	MNA	MNA	MNA	MNA	MNA
Id 10-18	MNA	MNA	MNA	MNA	MNA	Controls	MNA	Controls	MNA
Id 19-26	Controls	Controls	MNA	Controls	Controls	MNA	Controls	Controls	

Tabell 21: Klassificering obekräftat sjuka för bästa modell med data från **vad** för de 26 individer där data från båda mätområden finns tillgängligt.

Id 1-9	Controls	Controls	Controls	Controls	MNA	Controls	Controls	Controls	Controls
Id 10-18	Controls	MNA	Controls	Controls	Controls	Controls	Controls	Controls	Controls
Id 19-26	Controls	Controls	MNA	Controls	Controls	MNA	Controls	Controls	