



CHALMERS
UNIVERSITY OF TECHNOLOGY

Internet of things för optimerad och smart prestationsutveckling

Kandidatarbete inom civilingenjörsutbildningen
vid Chalmers tekniska högskola

Einar Ingemarsson

Thomas Trinh

Frida Krohn

Jessica Preiman

Sattar Akbari

Filip Helmroth

CHALMERS TEKNISKA HÖGSKOLA

Göteborg, Sverige 2021
www.chalmers.se

Internet of things för optimerad och smart prestationsutveckling

TIFX04-21-81

Einar Ingemarsson
Thomas Trinh
Frida Krohn
Jessica Preiman
Sattar Akbari
Filip Helmroth

ieinar@student.chalmers.se
ttrinh@student.chalmers.se
krohnf@student.chalmers.se
preiman@student.chalmers.se
sattara@student.chalmers.se
helmroth@student.chalmers.se

Handledare

Magnus Karlsteen
Ali Tabrizi
Janne Wanhainen
Jeremy Pryce
Darko Sarovic

magnus.karlsteen@chalmers.se
ali75tabrizi@gmail.com
janne.wanhainen@ifkgoteborgfriidrott.com
jeremy.pryce@ifkgoteborgfriidrott.com
darko@darkosarovic.com



CHALMERS
UNIVERSITY OF TECHNOLOGY

Institutionen för fysik
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2021

Internet of things för optimerad och smart prestationsutveckling
TIFX04-21-81

Einar Ingemarsson, Thomas Trinh, Frida Krohn, Jessica Preiman, Sattar Akbari, Filip Helmroth

© EINAR INGEMARSSON, THOMAS TRINH, FRIDA KROHN, JESSICA PREIMAN, SATTAR
AKBARI, FILIP HELMROTH, 2021.

Handledare: Magnus Karlsteen, Ali Tabriziali, Janne Wanhainen, Jeremy Pryce, Darko Sarovic
Examinator: Lena Falk, Institutionen för Fysik

Kandidatarbete 2021
Institutionen för fysik
Chalmers Tekniska Högskola
SE-412 96 Göteborg

Förord

Denna rapport är ett resultat av ett projektförslag av två tränare från IFK Göteborg friidrott: Jeremy Pryce och Janne Wanhainen. Vi vill tacka dem för deras idé och för deras fantastiska stöd och förtroende under arbetets gång. Vi vill även tacka dem för friheten vi fick att utveckla och styra riktningen på projektet åt det håll vi ansåg skulle vara mest till nytta för deras projektförslag.

Vi vill även tacka Ali Tabrizi som gett oss goda råd för utvecklandet av applikationen och som sett till att alla har ett leende på läpparna under handledningsmötena.

Vi vill även tacka Darko Sarovic för värdefulla reflektioner och idéer gällande teorin och studien, han har varit mycket hjälpsam från start och genom hela arbetet.

Till slut vill vi tacka Magnus Karlsteen som alltid ställde upp och svarade på våra frågor, både stora som små. Vi vill också tacka Magnus för det helhjärtade stödet och den positiva inställningen genom hela projektet.

Sammanfattning

Den växande marknaden kring digitala hjälpmedel som smarta klockor och pulsband visar på ett intresse för individuell träningsoptimering. Denna rapport syftade till att ta fram en grundläggande prototyp av en mobilapplikation med fokus på att intuitivt presentera analyserad träningsdata för slutanvändaren och en studie som jämförde prestationen mellan en matematisk modell och maskininlärningsmodeller. Projektet avgränsades till att använda existerande hårdvara och istället fokusera på analys av data. Studien avgränsades till att inte ta hänsyn till försökspersonernas näringsintag eller sömn, med undantag för antal sömntimmar. Applikationen strukturerades som en klientserverlösning där serversidan utvecklades som ett REST-API i Python med ramverket Flask och applikationssidan utvecklades med hjälp av react native. Studien genomfördes under 40 dagar med 6 försökspersoner med olika träningsbakgrund med ålder $23,2 \pm 1,6$ år. Samtliga skulle rapportera värden för rSPE varje dag och hade krav på att träna minst 4 gånger i veckan med varierad intensitet. Datan från studien bearbetades för att sedan användas i tre modeller: Banister-modellen och två maskininlärningsmodeller, en ANN-modell och en RNN-modell. MSPE användes för att mäta modellernas förmåga att förutsäga HRV. Medelvärde för MSPE \pm standardavvikelsen blev för Banister-modellen $0,066 \pm 0,052$, för ANN och RNN blev det $0,127 \pm 0,141$ respektive $0,05 \pm 0,005$. RNN påvisade ett tillräckligt lågt MSPE-värde utan att överanpassa datan och utsågs därför som ett lämpligt alternativ till Banister-modellen som fick det lägsta MSPE men istället överanpassade datan. Applikationen blev i huvudsak färdigställd men saknar en del önskad funktionalitet. Funktionalitet som blev implementerad både i mobilapplikationen, servern och databasen var: användarhantering med inloggning, inmatning av sRPE och HRV samt skapande av träningspass och uppvisning av förutsagt HRV. Applikationen funkar som en prototyp som kan vidareutvecklas i kombination med ytterligare studier med längre tidsintervall och fler försökspersoner för att vidare undersöka möjligheterna med att använda maskininläring för träningsoptimering.

Abstract

The growing market for digital aids such as smart watches and heart rate monitors shows an interest in individual training optimization. This report aimed to develop a basic prototype of a mobile application with a focus on intuitively presenting analyzed training data to the end user and a study comparing performance between a mathematical model and machine learning models. The project was limited to using existing hardware and instead focused on data analysis. The study was limited to not taking into account the subjects' nutritional intake or sleep, with the exception of the number of hours of sleep. The application was structured as a client server solution where the server side was developed as a REST API in Python with the framework Flask and the application side was developed using react native. The study was conducted over 40 days with 6 subjects with different training backgrounds aged 23.2 ± 1.6 years. All were to report values for rSPE every day and were required to exercise at least 4 times a week with varying intensity. The data from the study were processed and then used in three models: the Banister model and two machine learning models, an ANN model and an RNN model. MSPE was used to measure the models' ability to predict HRV. The mean value of the MSPE \pm standard deviation for the Banister model was 0.066 ± 0.052 , for ANN and RNN it was 0.127 ± 0.141 and 0.05 ± 0.005 , respectively. RNN demonstrated a sufficiently low MSPE value without over-adapting the data and was therefore designated as a suitable alternative to the Banister model, which received the lowest MSPE but instead over-adapted the data. The application was mainly completed but lacks some desired functionality. Functionality that was implemented in both the mobile application, the server and the database was: user management with login, input of sRPE and HRV as well as creation of training sessions and demonstration of predicted HRV. The application works as a prototype that can be further developed in combination with further studies with longer time intervals and more subjects to further investigate the possibilities of using machine learning for training optimization.

Innehållsförteckning

1	Inledning	1
1.1	Syfte	1
1.2	Avgränsningar	1
1.3	Marknadsanalys	1
1.4	Metodöversikt	2
2	Teori	2
2.1	Hjärtfrekvensvariabilitet	2
2.1.1	Faktorer som påverkar HRV	3
2.2	Kvantifiera träning	3
2.2.1	TRIMP	3
2.2.2	Session rating of perceived exertion	3
2.3	Analys	4
2.3.1	Banister-modellen	4
2.3.2	Artificiella neuronnät	5
2.3.3	Återkommande neuronnät	6
2.3.4	Jämföra modeller	7
3	Programvara och hjälpmedel	8
3.1	Git & GitHub	8
3.2	Figma	8
3.3	React Native	8
3.4	Expo	8
3.5	Flask	8
3.6	MongoDB	8
3.7	Elite HRV	9
3.8	Anaconda	9
3.9	Tensorflow	9
4	Applikation	9
4.1	Design	9
4.2	Server	11
4.2.1	Databas och API	12
4.2.2	Inloggningssystem	13
4.2.3	Dataanalys och maskininlärning i applikationen	13
4.2.4	Inhämtning av data via externt API	13
5	Metod för studie	13
5.1	Urval av försökspersoner	14
5.2	Mätning av HRV	14
5.3	Mätning av träningsbelastning	15
5.4	Databearbetning	15
5.4.1	Felinmatning av data	16
5.4.2	Sortering av data	17
5.4.3	Ofullständig data	17
5.4.4	Skalning av data	17
5.5	Regressionsanalys	17
5.5.1	Banister-modellen	17
5.5.2	ANN	18
5.5.3	RNN	19
6	Resultat	20
6.1	Jämförelse av modeller	20
6.2	Server	24
6.3	Applikation	24
7	Diskussion	25

7.1	Val av mätmetod	25
7.2	Val av värde att förutsäga	25
7.3	Tidsspann och datamängd	25
7.3.1	Urval av variabler i analysen	26
7.3.2	Framtida studier	26
7.4	Server och databas	26
7.5	Applikationens funktionalitet och vidareutveckling	27
7.6	Samhälleliga och etiska aspekter	28
Referenser		29
Appendix		31
A Applikationen		31
B Blankett för GDPR		39
C Enkäter för studien		40
D Resultat		42
D.1	Härledning	43

Ordlista

Bias: Bias är maskininlärningsmodellens tendens att konsekvent lära fel.

Dashboard: En dashboard är en visuell och lättbegriplig sammanställning av viktiga funktioner för att snabbt kunna få en översikt [1].

Dokumentdatabas: En databas som lagrar data i dokumentformat.

Fitness: Fitness används för att beskriva hur vältränad en individ är.

Hashfunktioner: En matematisk algoritm som genererar ett binärt värde med en given längd för indata med godtycklig längd. Vanligt förekommande när lösenord skickas över internet.

Homeostas: Homeostas är kroppens förmåga att hålla sig stabilt runt ett visst tillstånd trots förändringar i miljön. När man tränar utsätts kroppen för stress vilket rubbar den interna miljön. För att upprätthålla homeostasen anpassar sig kroppen genom att reglera den interna miljön. Viktiga homostatiska mekanismer är termoreglering, osmoreglering och reglering av blodsockernivåerna.

Högnivåspråk: Ett typ av programspråk som är mer abstrakt och där programmeraren inte behöver tänka på flyttinstruktioner eller minnesadresser.

Icke-relationsdatabas: En databas som sparar data på ett flexibelt sätt i motsats till en relationsdatabas där fokus ligger på att lagra relationer mellan data.

Overfitting: En maskininlärningsalgoritm som har uppnått för hög precision för träningsdata så att algoritmens tillförlitlighet på ny data är för låg.

Kodbas: All källkod som utgör grunden för att kompilera och bygga ett program eller en applikation.

Ramverk: Ett ramverk är ett färdigbyggt bibliotek av mjukvarukomponenter som kan användas till specifika syften.

Rutnätsökning: En teknik som försöker räkna ut det kommande värdet utifrån värden som redan blivit inmatade. När nya värden sedan matas in kontrollerar dessa att modellen gör rätt annars anpassar sig modellen på nytt [2].

Stressbiomarkör: En biomarkör är en mätbar biologisk parameter [3]. Detta innebär att en stressbiomarkör är en parameter som påverkas av stress. Träning usätter till exempel kroppen för en viss typ av stress.

Träningstimulans: En träningsstimulans är en form av stress som kroppen utsätts för vid fysisk aktivitet. Beroende på intensiteten och varaktigheten av stressen kan kroppen reagera genom till exempel muskelupbyggnad eller neuronal anpassning, vilket leder till en förbättrad fysisk form.

VO₂max: Maximum takt syre konsumerad medan träningen.

Akronymer

API: Application programming interface

ANN: Artificial neural network

ANS: Autonoma nervsystemet

AU: Arbitrary units

EMA: Exponential moving average

FPPT: Forward propagation through time

HRV: Heart rate variability

HTTP: Hypertext transfer protocol

JSON: JavaScript object notation

LSTM: Long short-time memory

MLPRegressor: Multi layer perceptron regressor

MSE: Mean squared error

MSPE: Mean squared prediction error

MVP: Minimum viable product. Avser den minsta möjliga produkten som fortfarande skapar värde för användaren.

NoSQL: Non structured query language

REST-API: Restful application programming interface.

RMSSD: Root mean square of the successive differences

RNN: Recurrent neural network

SQL: Structured query language

sRPE: Session rating of perceived exertion

TRIMP: Training impulse

1 Inledning

Historiskt har personlig utveckling inom sport och idrott varit enbart beroende av tränares erfarenheter och kunskap. På senare tid med den digitala utvecklingens framfart har det dock blivit allt mer vanligt att använda digitala hjälpmedel som till exempel smarta klockor. Dessa klockor har ofta möjlighet att mäta både puls och ge återkoppling kring träningens belastning samt frekvens. Oavsett om målet med träningen är hälsa eller prestation finns det ett intresse av att försöka optimera resultaten, vilket bekräftas av den växande marknaden för digitala hjälpmedel. En intressant frågeställning är då om man kan specificera informationen från både tränare och digitala hjälpmedel på ett sätt som är tillräckligt tillförlitligt.

För att ett resultat ska vara tillförlitligt krävs det att den underliggande datan har inhämtats med en hög noggrannhet och att den har en bevisad påverkan på resultatet. Det finns flera områden, bland annat vård och forskning, som idag är helt beroende av att arbeta med sådan data för att kunna dra tillförlitliga slutsatser. Metoder som utandningsanalyser, blodprov och hjärtfrekvens används regelbundet och i forskningens framkant upptäcks allt fler metoder och värden som kan vara till nytta för att göra ytterligare uttalanden. Med dessa värden som grund kan man med stora sannolikheter uttala sig om kroppens välmående. Därmed blir det även intressant ur ett träningsperspektiv att ta reda på om samma typer av värden kan ge betydelsefull information om träningsoptimering och skaderisker. För att kunna uttala sig om en individs välmående utifrån ett antal uppmätta värden krävs det ofta specialiserad utbildning, vilket kan vara rimligt inom områden som vård. När det istället gäller träningsoptimering vore det mer relevant om en vanlig idrottare kunde få ut samma typ av slutsatser med bibehållen korrekthet.

I dagsläget finns det många olika typer av redskap för att mäta viktiga värden vid träning. Genom mjukvara presenteras ofta mätvärden i tabeller och grafer, men värdet av dessa minskar utan specifik kunskap kring hur tabellerna och graferna ska tolkas i relation till ett specifikt mål. Idén till detta projekt uppstod i precis det tomrummet och syftar till att hitta en metod som kan ta dessa värden och presentera analysen av dessa på ett intuitivt sätt för varje individuell motionär.

1.1 Syfte

Syftet med projektet är att ta fram en grundläggande prototyp, en så kallad minimum viable product (MVP), i form av en mobilapplikation som kan hjälpa användaren att på ett mer individualiserat sätt få respons på sin träning. Detta genom att jämföra om modeller skapade med hjälp av maskininlärning kan förutsäga fysisk form utifrån träningsrelaterad indata bättre än konventionella matematiska modeller. Från resultatet av jämförelsen är syftet sedan att använda den bäst presterande modellen för att analysera användarens träningsdata och presentera detta på ett intuitivt sätt i applikationen. Målgruppen för projektet innefattar alla från vardagsmotionärer till elitidrottare.

1.2 Avgränsningar

Eftersom det redan finns ett flertal produkter som tillsammans täcker ett brett spektrum av mätvärden, avgränsas projektet till att använda befintlig hårdvara. Detta trots att projektet identifierat ett behov av ny hårdvara i form av ett bärbart system för mätning av utandningsluft som kan bäras bekvämt under ett träningspass. Beslutet togs med hänsyn till att utvecklandet av en sådan produkt med stor sannolikhet skulle bli tidskrävande. Dessutom fastslogs att ett grundläggande system med fokus på dataanalys och användargränssnitt är mer värdeskapande för en MVP än en prototyp av ovannämnda hårdvara.

Sömn och näringsintag har stor inverkan på en persons energinivå och prestationsförmåga. Dock har avgränsningar gjorts där analysen kommer samla in subjektiv data i form av sömntimmar men bortse från all data kring näringsintag. Detta innebär att användaren svarar på frågor kring sin sömn istället för att genomföra objektiva mätningar. Valet att bortse från näringsintag grundar sig i att datan kräver detaljerad manuell inmatning från användaren, vilket sannolikt är för tidskrävande i förhållande till dess påverkan på analysen. Till skillnad från näringsintaget, kan mängden sömn och dess kvalitet lättare uppskattas samt matas in av användaren.

1.3 Marknadsanalys

Redan idag finns det flera applikationer som uppfyller liknande syfte som detta projekt vill uppnå. Tjänsterna Strava, Runkeeper och Svetlig är tre vanliga träningsapplikationer vars främsta syfte är att skapa en social gemenskap kring träningen där man kan logga och dela sina träningspass med bekanta.

Tjänsterna Polar Flow, Garmin Connect, EliteHRV och Apple Health är fyra applikationer som fokuserar mer på analysering av träningsdata och hur träning i kombination med återhämtning kan optimeras för att producera bättre träningsresultat. Applikationen ämnar att bli en tjänst som både inkluderar den sociala gemenskapen och den individuella prestationsutveckling. Därmed skulle den fylla ett identifierat tomrum på marknaden.

1.4 Metodöversikt

Detta projekt har genomförts i två huvudsakliga delar: en studie och utveckling av en applikation. Strukturen på rapporten kommer därför beskrivas kort för att tydliggöra rapportens kapiteluppdelning. Först presenteras teorin som ligger till grund för studien. Detta efterföljs av programvara och hjälpmedel som beskriver verktygsval för utvecklandet av applikationen och analysverktyget. Utvecklingsprocessen för applikationen beskrivs sedan i sin helhet. Metoden för genomförandet av studien presenteras därefter och avslutningsvis presenteras resultat för både studien och utveckling av applikationen samt en diskussion kring bägge dessa resultat.

2 Teori

En stor utmaning är att hitta balansen mellan träning och vila över längre tidsperioder där prestationsutvecklingen maximeras och risken för överträning och skador samtidigt minimeras. Ett sätt att angripa detta problem är att kartlägga en idrottares fitness och analysera hur den förändras över tid. Utifrån dessa förändringar går det sedan att se om idrottaren får positiv eller negativ effekt på sin utveckling utifrån sitt aktuella träningsprogram [4]. Genom att undersöka effekterna kan det exempelvis avgöras om idrottaren kan träna mer intensivt, vilket leder till en förbättrad prestationsutveckling, eller om idrottaren behöver mer vila, då en för hög träningsbelastning kan leda till en försämrad prestationsutveckling och dessutom en högre risk för skador [5].

Det vanligaste sättet att kartlägga en idrottares fysiska form är att undersöka de olika typer av träningsbelastning som idrottaren utsätts för under sitt träningsprogram. Träningsbelastning kan kategoriseras som antingen intern eller extern belastning [4]. Intern träningsbelastning omfattar alla stressbiomarkörer som uppstår vid fysisk ansträngning hos idrottare, både fysiologiska och psykologiska. En stressbiomarkör är en mätbar biologisk parameter som påverkas av stress och några exempel på stressbiomarkörer som är vanliga att mäta är puls, hjärtfrekvensvariabilitet och subjektiv upplevd fysisk ansträngning. Extern träningsbelastning avser istället objektiva mätningar som utförs på den aktivitet idrottaren genomför. Detta inkluderar exempelvis mätning av acceleration, hastighet, kraft och träningseffekt.

2.1 Hjärtfrekvensvariabilitet

Hjärtfrekvensvariabilitet (HRV) är en av de stressbiomarkörer som går att mäta för att få en uppfattning av den interna träningsbelastningen. Värdet erhålls genom att mäta variationen mellan hjärtslagens tidsintervall [6]. HRV påverkas av det autonoma nervsystemet (ANS) som reglerar kroppens grundläggande livsprocesser, till exempel puls, andning och matsmältning. ANS är uppdelat i två delar: det sympatiska och parasympatiska nervsystemet [7]. Det sympatiska nervsystemet styr energikrävande uppgifter medan det parasympatiska styr över energisparande uppgifter. Dessa två system motarbetar alltså varandra där större delen av det sympatiska systemet är igång när du exempelvis tränar medan det parasympatiska är igång när du vilar och återhämtar dig. ANS:s primära funktion är att alltid försöka behålla homeostas genom reglering av olika kroppsfunktioner. När en idrottare utsätter sin kropp för träningsstress under upprepade träningspass leder det till fysiologisk anpassning, vilket innebär att den homeostatiska basnivån reduceras som respons. Följaktligen kan ANS:s respons till ändring i träningsbelastning indikera kroppens förmåga att tolerera eller anpassa sig till träningsstimulus [8].

HRV kan mätas med ett pulsband som är ett kostnadseffektivt, icke-invasivt och praktiskt sätt att uppskatta ANS reglering av pulsvariabiliteten. Värdet på mätningen representerar summan av effekterna på det sympatiska och parasympatiska systemet i respons till både fysisk och psykologisk stimulus [9]. Detta innebär att det sympatiska nervsystemet aktiveras när en individ utsätts för fysisk ansträngning, vilket leder till en förhöjd puls och ett minskat HRV. När kroppen däremot får vila aktiveras det parasympatiska systemet, vilket leder till en sänkt puls och ett högre HRV [10]. Trots att det finns tydliga fysiska och

fysiologiska skillnader mellan idrottsutövare, ökar för närvarande användandet av HRV-mätningar inom idrottsvetenskap för att bevaka idrottare [11].

2.1.1 Faktorer som påverkar HRV

Vid användning av HRV som ett mätvärde är det viktigt att fastställa vilka yttre faktorer som kan påverka mätningen. Detta är framförallt viktigt när man ska analysera och dra vidare slutsatser utifrån dessa värden genom att finna underliggande mönster samt förklara avvikelser i den insamlade datan. Några kända faktorer som påverkar HRV, men som en individ själv inte kan påverka är ålder, kön och dygnsrytm. En individs HRV ökar fram tills cirka 15 års ålder och börjar sedan avta med åldern.

Man har observerat att försökspersoner i medelåldern följer HRV en periodisk dygnsrytm [6]. Observationer har visat att HRV-värdet minskar under förmiddagen och antar sina lägsta värden på eftermiddagen runt 15:00-18:00 för att sedan öka igen på kvällen och antar sina högsta värden tidigt på morgonen runt 3:00-5:00.

HRV kan också påverkas av vilken miljö individiden befinner sig i. Detta beror på att miljöfaktorer kan leda till psykologisk stimulans vilket påverkar ANS. Exempelvis kan exponering för ljud leda till att det sympatiska nervsystemets aktivitet ökar och därmed sjunker HRV-värdet [12]. På samma sätt kan värme sänka HRV-värdet genom en ökning av det sympatiska systemets aktivitet.

2.2 Kvantifiera träning

För att få ett pålitligt värde på idrottarens träningsbelastning är det viktigt att använda sig av flera olika mätvärden då enskilda värden riskerar att ge en felaktig bild. För att ytterligare addera värde kan sedan varje värdes påverkan på belastningen undersökas. Träningsbelastning påverkas huvudsakligen av tre komponenter: varaktighet, träningsintensitet och träningsfrekvens [13]. Summan av dessa komponenter kan beskrivas som en träningsstimulans. När det kommer till förbättring av prestationsförmåga är det dock svårt att definiera gränsen mellan optimal och destruktiv träningsstimulans.

2.2.1 TRIMP

Ett av de mer populära sätten att kvantifiera träningsbelastning är att mäta hjärtfrekvensen hos en idrottare under ett träningspass. Utifrån dessa värden kan sedan TRIMP-metoden användas för att räkna ut träningsbelastningen genom att ta varaktigheten multiplicerat med förändringen i hjärtfrekvens [14]. Förändringar i hjärtfrekvens uppstår då intensiteten förändras under ett träningspass, där förväntningen är en högre hjärtfrekvens vid hög träningsintensitet och en lägre hjärtfrekvens vid låg träningsintensitet.

Genom att använda kontinuerlig pulldata som sträcker sig över ett träningspass kan dessa värden sedan matas in i ekvationen och producera en kvantifierad träningsbelastning. Detta då hjärtfrekvens kan ses som ett index på utnyttjandet av sin maximala syrekonsumtion.

Träningsbelastningen kan beräknas vid varje tidpunkt, t , genom att ta produkten av ett träningsmoments varaktighet och förändring i hjärtfrekvens vid tiden t . Träningsbelastningen vid tiden t , kan därmed representeras av pseudointegralen av ekvation (1).

$$w(t) = (\text{varaktigheten av träningen}) \cdot \frac{HR_{\text{ex}} - HR_{\text{vila}}}{HR_{\text{max}} - HR_{\text{vila}}}, \quad (1)$$

där HR_{ex} är medelpulsen, HR_{vila} är vilopulsen och HR_{max} är maxpulsen.

2.2.2 Session rating of perceived exertion

Session rating of perceived exertion (sRPE) är en metod för att mäta träningsbelastning där idrottaren får ange upplevda värden på träningsanstränging. Denna metod, uppfunnen av Carl Foster, låter idrottare helhetsgradera ansträngingen av träningspasset 30 min efter passets slut [15]. Det ska helst inte graderas tidigare än 30 min och inte heller långt senare. Om träningsintensiteten graderas direkt efter träningen finns en risk att graderingen endast reflekterar slutet av passet och inte hela passet. Att graderingen inte bör ske för sent beror på att en förlängd vila efter passet kan påverka graderingen också. Därav ses 30 min som en lämplig tid för att graderingen ska reflektera hela passet [16]. Graderingen utförs utifrån en skala mellan 6 och 20 som kan ses i tabell 1. Graderingen av passet, RPE, multipliceras sedan med

varaktigheten av träningen vilket ger ett värde på träningsbelastningen i en arbiträr enhet, se ekvation (2).

$$\text{Träningsbelastning} = \text{RPE} \cdot (\text{varaktighet i minuter}) \quad (2)$$

Gradering	Ansträngning
6	Extremt lätt
7	Extremt lätt
8	Extremt lätt
9	Mycket lätt
10	Mycket lätt
11	Lätt
12	Lätt
13	Ganska ansträngande
14	Ganska ansträngande
15	Ansträngande
16	Ansträngande
17	Mycket ansträngande
18	Mycket ansträngande
19	Extremt ansträngande
20	Maximalt ansträngande

Tabell 1: Tabell av Borgs RPE-skala.

2.3 Analys

Data kan analyseras med hjälp av olika matematiska modeller med förhoppning att finna samband. För att hitta samband kan man utifrån valda parametrar använda sig av färdiga matematiska modeller eller använda maskininlärning som letar efter matematiska samband som sedan kontinuerligt förbättras i takt med att datamängden ökar.

2.3.1 Banister-modellen

Vid modellering av prestationsförmåga finns endast ett fåtal färdiga modeller som är användbara. En av dessa är impuls-responsmodellen som är uppkallad efter skaparen Banister. Denna har visat sig kunna förutsäga träningsinducerade träningsresponser på prestationsförmåga med hyfsad exakthet i diverse uthållighetssporter och icke-uthållighetssporter [17]. Utifrån modellen har sedan förslag på ideala träningsprogram utarbetats där målet varit att optimera förhållandet mellan träning och vila. Banister-modellen modellerar relationen mellan träning och prestationsförmåga som en funktion. Indata representeras av träningsstimulanser och utdata representerar idrottarens uppskattade prestation [17].

Funktionen beskrivs med hjälp av två faktorer $g(t)$ och $h(t)$, se ekvation (3) respektive ekvation (4).

$$g(t) = g(t - i)e^{-i/\tau_1} + w(t) \quad (3)$$

$$h(t) = h(t - i)e^{-i/\tau_2} + w(t) \quad (4)$$

Dessa ekvationer motsvarar fitness och utmattning för dagen t . $g(t)$ representerar mer långvariga och positiva effekter från träningsstimulanser medan $h(t)$ representerar kortvariga och negativa effekter av träningen. $w(t)$ är träningsbelastning, i är antalet vilodagar och τ_1 samt τ_2 är tidskonstanter som påverkar hur snabbt funktionerna avtar.

Eftersom fitnessfaktorn har en positiv påverkan och utmattningsfaktorn har en negativ påverkan på prestation, kan man modellera prestationsutveckling med den linjära differensen i ekvation (5).

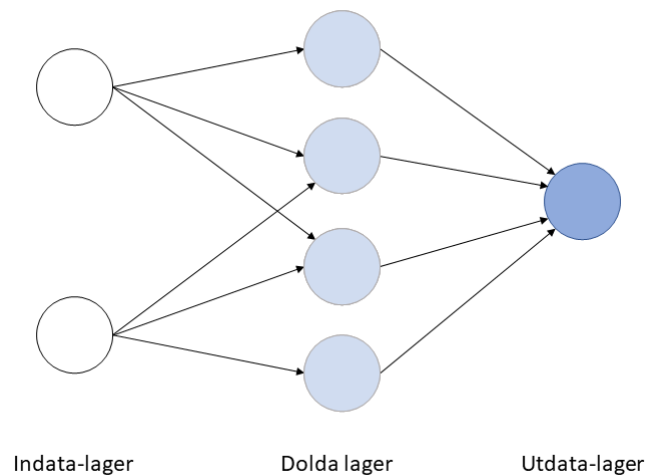
$$p(t) = k_1 g(t) - k_2 h(t), \quad (5)$$

där $p(t)$ är prestationen vid tiden t och k_1 samt k_2 är positiva dimensionlösa viktkonstanter som är relaterade till effekterna av respektive faktor. Idrottare med ett större k_2 karaktäriseras av en utmattningsdominerad prestationsutveckling, vilket innebär att det tar längre tid för dem att återhämta sig från intensiv träning [17]. Idrottare med ett större k_1 värde har däremot en fitnessdominerad prestationsutveckling vilket betyder att de återhämtar sig snabbare.

2.3.2 Artificiella neuronät

Ett artificiellt neuronät (ANN) är ett koncept som är inspirerat av biologiska neuronät i hjärnan, där flera självlärande algoritmer har som uppgift att i en mängd data leta efter mönster och samband [18]. Nätet består av ett flertal sammankopplade noder som ska representera hjärnans uppbyggnad av neuroner. Från ett matematiskt perspektiv är ett neuronät en matematisk funktion som tar in och klassificerar information enligt ett specifikt nätverk [19]. Processen i ANN är väldigt lik processen i statistiska metoder som regressionsanalys och interpolation.

Noderna i ett neuronät delas in i flera olika lager där varje nod har en anpassningsbar parameter som kallas för en vikt. Det första lagret tar in obehandlad data som viktas för att skickas vidare till nästa lager som är dolt. Denna process upprepas sedan lika många gånger som det finns dolda lager för att till sist resultera i ett eller flera slutvärden som utdata. På detta sätt bildar noderna ett nätverk som innehåller nodernas relation till varandra, se figur 1.



Figur 1: Ett enkelt neuronät med två noder för indata, fyra noder i ett dolt lager och en nod för utdata.

Värdet i varje nod kan beräknas genom en linjärkombination där x_i representerar individuella värden från noder i det tidigare lagret med tillhörande vikt w_i [20]. Vanligtvis justeras sedan detta värde med hjälp av en aktiveringsfunktion f för att bestämma hur mycket information som ska skickas till nästa nod. Exempel på aktiveringsfunktioner är identitetsfunktionen, stegfunktionen och sigmoidfunktionen. Dessa två steg tillsammans kallas ofta för framåtpropagering, se ekvation (6).

$$y(n) = f\left(\sum_{i=1}^n (x_i \cdot w_i)\right) = f((x_1 \cdot w_1) + (x_2 \cdot w_2) + \dots + (x_n \cdot w_n)), \quad (6)$$

där x_i är värdet från en nod i det tidigare lagret, w_i är tillhörande vikt och f står för aktiveringsfunktionen. Noderna i det sista lagret utgör resultatet och kan användas för att dra slutsatser om kopplingar i datamängden, till exempel bildigenkänning eller att förutsäga värden utifrån en viss indata.

Nätverket tränas sedan utifrån ett dataset där vikterna justeras utifrån hur nära slutvärden stämde med det faktiska värdet från datasetet [20]. Funktionen för att räkna ut felgraden i slutvärden kan definieras

som ekvation (7).

$$E(n) = \frac{1}{2} \sum_j e_j^2(n) = \frac{1}{2} \sum_j ((d_j(n) - y_j(n))^2 \quad (7)$$

Där j är noderna i det sista lagret i datapunkten n med $e_j(n) = d_j(n) - y_j(n)$, d är målvärdet och y är nodens värde som beräknas i (6).

För att minska värdet på felgraden används en process som kallas bakåtpropagering där gradienten dE/dw_i av felgraden E beräknas och uppdateras för varje vikt w_i och η är träningsstakten för modellen, se ekvation (8).

$$w_{i,new} = w_{i,old} - \eta \frac{dE}{dw_i}. \quad (8)$$

När värdet på denna funktion anses vara tillräckligt minimerat kan nätverket ses som tränat.

2.3.3 Återkommande neuronnät

Återkommande neuronnät (RNN) är en typ av ANN som lämpar sig för att hitta mönster i sekventiell data, som till exempel text och tidsserier [21]. Detta åstadkoms genom att varje nod i RNN-modellen inte bara får indata från tidigare lager, utan även återkoppling från tidigare iterationer i beräkningsstegen. Således kan modellen lära sig hur sekventiella värden samverkar. Detta kan jämföras med ett vanligt neuralt nätverk där ingen återkoppling sker [21].

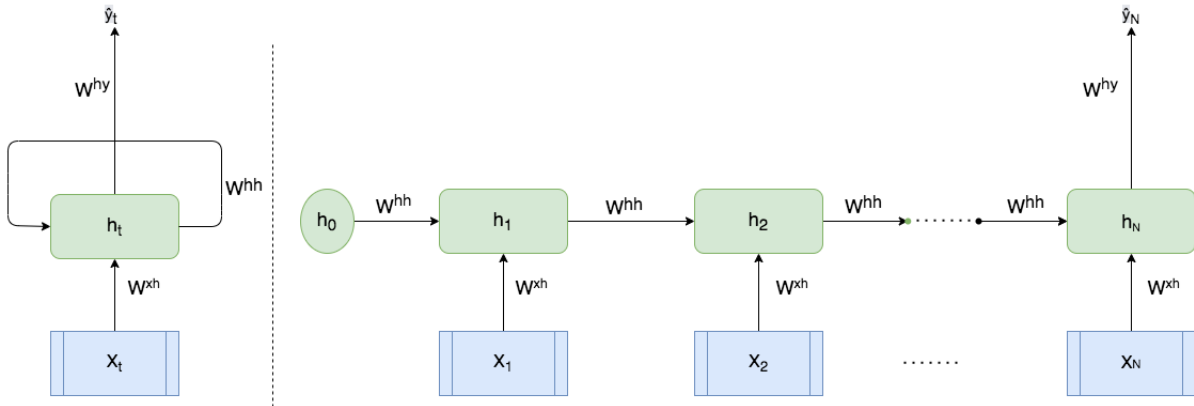


Figure 2: En återkommande neuronnät med N tidenheter

Arkitekturen för en RNN-modell visas i figur 2 [22]. Under ett tidssteg, t , tas indatavektorn X_t samt det tidigare tillståndet h_{t-1} och skapar det nya tillståndet h_t genom aktiveringsfunktionen (Φ). Detta visas i ekvation (9) [23]. Vid tillstånd 0, h_0 , existerar det inget tillstånd än och således initieras tillståndet med en tom vektor. Utdatan från RNN-modellen, y_N , fås genom ekvationen (10) och sker via framåtpropagering [23]. Denna typ av RNN-modell klassificeras som en *many-to-one*, vilket innebär att indata består av flera tidsobservationer medan utdata endast erhålls i den sista tidpunkten [24].

$$h_t = f(x_t, h_{t-1}) = \Phi(W^{xh}x_t + W^{hh}h_{t-1} + b^h) \quad (9)$$

$$y_t = W^{hy}h_t + b^y \quad (10)$$

I ekvationerna 9 och 10 finns även två biases, nämligen b^h och b^y . Ett bias är ytterligare en parameter som uppdateras medans neuronnätet kalibreras. En RNN-modell har tre olika samlingar av vikter: W^{xh} , W^{hh} och W^{hy} , som kan ses i ekvation (9) och (10). Dessa vikter representerar olika tillstånd och fungerar som filter för att bestämma hur stor vikt indata (x_t) respektive tillståndet (h_{t-1}) ska få. Felet som genereras justeras genom bakåtpropagering vid varje iteration. En iteration innebär att algoritmen går

igenom och justerar alla vikter en gång, vilket leder till att felmarginalen minskas [24]. Felet beräknas genom en förlustfunktion, L , enligt ekvation (11) [21].

$$L = \sum_{i=0}^N l_i = \sum_{i=0}^N f(y_i, \hat{y}_i) \quad (11)$$

Förlustfunktionen kan anpassas till RNN-modellens syfte, och är en funktion av y och \hat{y} . Dessa justeras via bakåtpropagering där algoritmen stegvis går tillbaka genom tidsenheterna och justerar vikterna. Följaktligen blir RNN-modellen mer och mer kalibrerad och felmarginalen minskar [21]. Ett vanligt återkommande problem med RNN benämns *vanishing gradient och exploding gradient* [21]. I detta fall syftar gradienten på den term som uppdaterar vikten W^{hh} , vilket kan ses i figur 2 och visas i ekvation (12) [24].

$$\nabla = \frac{\partial L(\hat{y}, y)}{\partial W} \quad (12)$$

Gradienten definieras som kvoten mellan förändringen i felmarginal genom förändring av vikten i föregående beräkningssteg. Blir gradienten för stor resulterar det i att vikterna uppdateras för snabbt. Blir gradienten för liten riskerar vikterna att uppdateras för långsamt [21]. En vedertagen lösning till problemet är *Long Short Time Memory* (LSTM), vilket är en typ av RNN [25]. Arkitekturen på en LSTM-modell är väsentligt mer komplex där man lägger till ytterligare en tredje tillstånds-input [21].

2.3.4 Jämföra modeller

För att avgöra hur bra en matematisk modell presterar krävs att en uppsättning av data delas upp i en träningsuppsättning och en testuppsättning. Träningsdata används för att träna modellen medan testdata används för att utvärdera modellens prestation. Vanliga delningsförhållanden är 80/20, 67/33, 50/50. Där 80/20 betyder att 80% av datan används för uppträning och 20% av datan används för testning.

Vid utvärdering av modeller kan måttet mean squared prediction error (MSPE) användas, vilket mäter modellernas förmåga att förutsäga rätt värde. Idealt ska värdet vara noll vilket innebär att modellen har lyckats förutsäga de verkliga värdena. MSPE beräknas med ekvation (13).

$$\text{MSPE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (13)$$

En annat mått som kan användas för att jämföra modeller är korrelationskoefficient r [26]. Detta måttet kan appliceras på träningsperioden för att kvantifiera hur bra modellen passar träningsdatan. Koefficienten antar ett värde mellan -1.0 och $+1.0$. Ett r -värde på $+1$ innebär att datan passar perfekt medan -1 har en inverterad korrelation.

3 Programvara och hjälpmedel

Flera olika programvaror och hjälpmedel användes under arbetets gång för att möjliggöra och förenkla olika processer samt för att underlätta samarbete. Samtliga av dessa beskrivs kort tillsammans med en förklaring till varför de behövdes.

3.1 Git & GitHub

För att slippa spara kod manuellt och hålla reda på vilken version som är rätt används Git som är ett verktyg för versionshantering. Med hjälp av Git kan man se till att all kodhistorik dokumenteras och sparas. Detta innefattar även vem som har skrivit eller tagit bort specifika kodrader i kodbasen. När Git används i kombination med Github, som är en online-plattform för att lagra kod, kan kodbasen lagras i molnet och därmed förenklas processen att dela kod mellan programmerare.

3.2 Figma

Som prototypverktyg och grafikredigeringsprogram valdes Figma. Bakgrunden till valet var Figma:s möjligheter att samarbeta med andra personer i realtid, att skapa interaktiva prototyper och att dessa kan köras i en mobiltelefonsimulator. I prototypen kan man även navigera mellan de olika sidorna som om den vore en färdig applikation. Detta underlättar nästa projektfas, då applikationen ska programmeras, eftersom det finns en klar bild över vad som ska åstadkommas.

3.3 React Native

Ramverket som valdes var React Native och använder sig av programmeringsspråket JavaScript. React native är anpassat för att förenkla processen att utveckla mobilapplikationer. För att skapa en lättillgänglig applikation bör den finnas tillgänglig för både iOS- och Androidanvändare, vilket React Native tar hand om genom att agera som en egen plattform som internt hanterar kodomvandlingen till respektive operativsystem. Detta innebär att kodbasen därför bara behöver skrivas en gång i ramverkets språk. Detta blev extra viktigt då projektet var tidsbegränsat och det därför inte fanns möjlighet att skapa två separata applikationer.

3.4 Expo

Expo är ett ramverk som gör det enkelt att testa applikationen via både webbläsaren och mobiltelefonen, där det senare möjliggörs via en specifik app: Expo Go. Vidare så möjliggör expo att applikationen uppdateras i realtid samtidigt som programmeringskoden ändras utan att applikationen i sig behöver startas om. Syftet med att använda expo var att underlätta och öka hastigheten på skapandet av applikationen.

3.5 Flask

Flask är ett ramverk som förenklar programmeringen av servern. Flask är skrivet för programmeringsspråket Python och underlättar utvecklingen av serverapplikationen och specifikt hur API:t kan struktureras. Programmeringsspråket Python är ett väldokumenterat högnivåspråk som har ett stort utbud av standardiserade funktioner och bibliotek. Ramverket Flask fungerar väl att integrera mot React Native, vilket beskrivs i avsnitt 3.3.

3.6 MongoDB

MongoDB är en icke-relationsdatabas, så kallad non Structured Query Language (NoSQL), vilken lagrar data på ett sätt som är optimerat för de specifika kraven som applikationen har. Denna typ av databas möjliggör skalbarhet, smidig utveckling och hög flexibilitet jämfört med en traditionell relationsdatabas, Structured Query Language (SQL). Skalbarheten var en av nyckelfaktorerna till valet av denna modell. Genom att från start använda en skalbar databas möjliggörs att applikationen i framtiden enkelt kan addera nya datastrukturer utan att behöva bygga om databasen. På grund av säkerhetsskäl, vilket vi återkommer till i avsnitt 4.2.2, valde vi att bygga två separerade databaser för användaruppgifter och hälsodata.

Databasen implementeras genom applikationen MongoDB Atlas vilket är en extern molnbaserad dokumentdatabas som kan driftsättas på valfri molnplattform. MongoDB har inbyggd funktionalitet för att strukturera upp data effektivt och optimera arbetsbelastningen. Den skapar även en flexibilitet då den kan hantera multipla förfrågningar parallellt, vilket är fördelaktigt då flera användare ska kunna använda applikationen samtidigt.

3.7 Elite HRV

Elite HRV är en applikation som mäter och analyserar hjärtvariabiliteten. Applikationen är kompatibel med flera befintliga pulssensorer, vilket förenklar mätningar och uppföljning av dagliga HRV-mätningar till studien. Dessa mätningar är en förutsättning för att kunna göra de analyser som beskrivs i avsnitt 5.2.

3.8 Anaconda

Anaconda är en komplett plattform för dataanalys i programmeringsspråket Python. Anaconda innehåller flera förinstallerade programvarubibliotek som används för dataanalys och statistik. Programvarubiblioteken som användes från Anaconda var främst NumPy, Scikit-Learn, Pandas, SciPy och Matplotlib.

3.9 Tensorflow

Tensorflow är ett heltäckande programvarubibliotek för maskininlärning som är skapad av Google. Tensorflow erbjuder ett programvarubibliotek som kan träna och köra neurala nätverk vilket används i projektet. Keras är en modul som körs ovanpå Tensorflow och bidrar med ett användarvänligt gränssnitt för att kunna arbeta med programvarubiblioteket Tensorflow.

4 Applikation

För att kunna utnyttja studien och dess resultat behövs verktyg för att lagra resultaten från analysen, samt ett sätt att redovisa den för användaren. Därför väljer detta projekt att även fokusera på att skapa en server och en applikation för att kunna kommunicera med användaren.

Applikationens syfte är att ge användaren ett lätt och enkelt sätt att få översikt över sin träning och prestationsutveckling. För att åstadkomma detta bör applikationen vara intuitiv att hantera och förstå, den ska alltså kunna visa resultatet från analysen på ett sätt som är lätt att tyda. I applikationen ska det finnas möjlighet att starta och logga träningspass för att efteråt visa lämplig feedback samt en funktion att kunna se tidigare träningspass. I applikationen kommer även användarens subjektiva data tas in.

4.1 Design

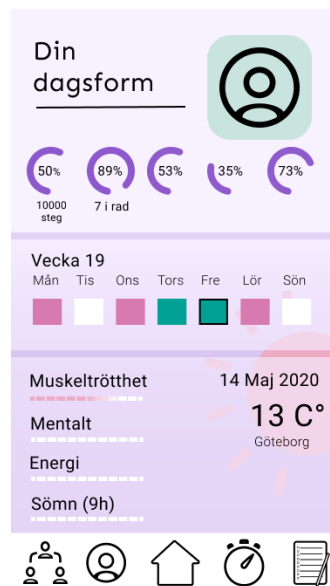
För att skapa en intuitiv design för applikationen krävs att användarflödet är väl genomtänkt. Den första planeringen gjordes helt i text och fungerade som en utgångspunkt för designen. Dess syfte var att logiskt kunna sortera alla önskade funktioner så att designen kunde utformas därefter, detta både för att göra applikationen lättnavigerad samt se till att den uppfyller önskad funktion. I steget efter skapades en prototyp av applikationen i Figma vilket gav en tydlig bild över designen och underlättade skapandefasen. Ännu en fördel med att ha en prototyp var att skapa en större medvetenhet kring designen. När förståelsen ökar kring navigation i applikationen kan den skapas på ett sätt där den beter sig så som användaren förväntar sig.

Applikationen delades in i fem huvudsidor: dashboard, aktivitet, historik, profil samt en sida för grupper och organisationer. Sidorna kodades separat och kopplades sedan till varandra via en rad ikoner som fungerar som applikationens huvudmeny.

Applikationens dashboard ses i figur 3. Dashboarden finns till för att användaren ska ha en god översikt på sin träningshistorik, subjektiva data och mål. Utöver detta kan användaren se dagens datum, vad vädret är just nu och fylla i sin subjektiva data.

Nya träningspass startas på aktivitetssidan som ses i figur 4, där finns möjlighet att välja vilka hårdvaror man vill använda och ansluta dessa via bluetooth. När en aktivitet sedan startas visas ett nytt fönster

med en knapp där användaren kan pausa respektive fortsätta passet. Utöver detta går det även att avsluta passet samt se ett tidtagarur med träningens varaktighet.



Figur 3: Figma-prototypens dashboard.



Figur 4: Figma-prototypens sida för pågående aktivitet.

Historiksidan delas upp i tre flikar, ett flöde, en översikt och tidigare värden. Flödet visar en lista med information över de tidigare passen så som dess givna namn, tid och datum för utförande. Även data från eventuell kopplad hårdvara kan visas, så som genomsnittligt tempo eller snittpuls. Historikflödet kan ses i figur 5. Översikten har en månadskalender som markerar de dagar där aktiviteter har genomförts. Under samma flik finns även statistik för månaden som visar antal loggade pass, loggade timmar och hur många dagar användaren uppfyllt respektive mål. Under tidigare värden kommer bland annat HRV för de olika dagarna synas.

Profilsidan, som syns i figur 6, finns då användaren behöver ha en profil för att applikationens server ska kunna lagra information om användaren. På denna sida kan användaren ställa in sina mål, lägga in en profilbild, redigera kontoinformation samt både importera och exportera data till respektive från applikationen.

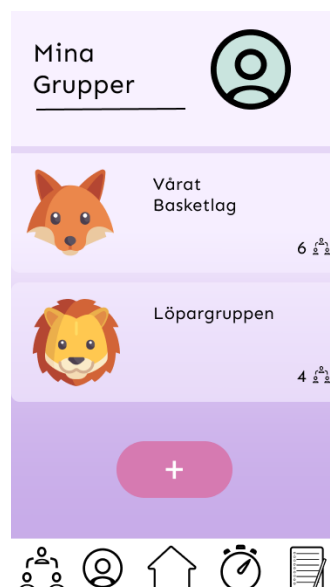


Figur 5: Figma-prototypens historikflöde.



Figur 6: Figma-prototypens profilsida.

På gruppssidan finns möjligheten att skapa en ny grupp och se alla grupper användaren är med i, denna sida syns i figur 7. Det finns två olika typer av grupper: kompisgrupp och organisation. En kompisgrupp är till för att användare ska kunna dela sin träning med varandra och även kunna sätta gemensamma mål, hur en sådan kan se ut syns i figur 8. En organisation är till för att tränare för olika idrottslag och träningsgrupper ska kunna följa sina adepters träning och värden. Skillnaderna mellan dessa grupper är att i en kompisgrupp kan alla se varandras träningspass och värden, samt deras individuella bidrag till gruppmålen. I en organisation däremot är de bara tränaren som ser träningspassen och värdena, samt kan ändra inställningarna i gruppen. Adepterna i en organisation kan se vilka som är med i gruppen och se sitt personliga träningsprogram som tränaren skrivit in.



Figur 7: Figma-prototypens gruppssida.

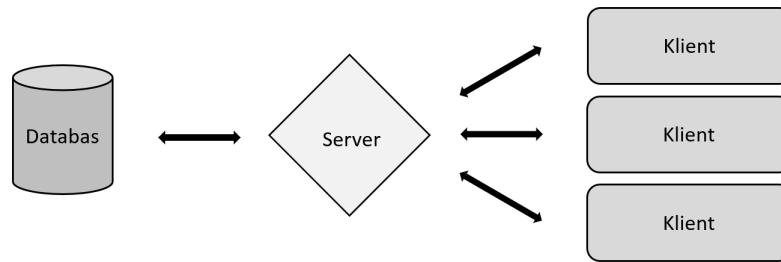


Figur 8: Figma-prototypens sida då en grupp valts.

4.2 Server

Systemet implementerades med en klientserverlösning enligt figur 9. Klienterna kommunicerar med servern genom API-anrop. Servern kommunicerar sedan med den externa databasen där all information

ligger lagrad. Denna arkitektur ansågs skapa en bra skalbarhet för applikationen och är en väl beprövad arkitektur.



Figur 9: Figuren visar den övergripande systemarkitekturen och hur de olika komponenterna kommunicerar.

Utvecklingen av servern skedde parallellt med klientsidans mobilapplikation. Fokus för servern låg på att implementera de mest systemkritiska funktionerna, vilket innefattade databashantering och utveckling av standardiserade API-anrop. Därefter fortskred arbetet med att sammankoppla klientsidan med servern, dataprocessering samt att inhämta träningsdata från externa tjänster likt Strava. Tjänsten Strava förklaras mer utförligt i avsnitt 4.2.4.

4.2.1 Databas och API

Servern använder sig av en icke-relationsdatabas där fördelarna har beskrivits utförligt i avsnitt 3.6. Databasen lagrar dokument i JSON-format vilket är lätthanterligt både för servern och klienten. Detta formatet visualiseras i figur 10 och kan förklaras som ett nyckel-till-värde-par. Varje textnyckel refererar till ett specifikt värde. Formatet gör att det är snabbt och enkelt att extrahera specifik data ur databasen.

Databasen strukturerades i två övergripande kategorier: hälsodata och användaruppgifter. Hälsodatan består av både individuell träningsdata, HRV och upplevd träningsbelastning. Bakgrunden till den inhämtade datan beskrivs i avsnitt 5.2 och 5.3. Figur 10 visar hur ett datainsamlingsformulär är lagrat i databasen. Användaruppgifterna som lagras i databasen har två syften: Dels att lagra vilka användare som finns i systemet, men också för att ge korrekt behörighet till hälsodatan som finns lagrad. Dessa syften ledde till att vi valde att implementera ett inloggningssystem, vilket beskrivs mer utförligt i avsnitt 4.2.2.

```
1  {
2      "stress_level": 3,
3      "muscle_ache": 4,
4      "mood_level": 10,
5      "hrv": 84,
6      "injury_level": 2,
7      "date": "2021-04-01",
8      "energy_level": 5,
9      "sleeping_hours": 8
10 }
```

Figur 10: Exempel på en JSON-fil som sparar en användares "mätning av hjärtfrekvensvariabilitet" enkät i databasen.

För att möjliggöra enkel och flexibel kommunikation mellan servern och flera klienter parallellt beslutades det att ett API skulle användas. Detta var en förutsättning för att slutprodukten i framtiden ska kunna fungera över internet och att användarna ska kunna utnyttja applikationen via sina personliga mobila enheter. Valet föll på att implementera ett så kallat REST-API. I praktiken betyder det att klienten är separerad från servern och datalagringen, vilket innebär att servern inte vet vad som sker hos klienten och vice versa. Ett REST-API tillåter klienten att hämta och manipulera specifik data i databasen genom standardiserade API-anrop. Hur ett API-anrop implementeras kan ses i tabell 2. Ramverket Flask, som

nämndes i avsnitt 3.5, används för att hantera anropen från klienten. De anrop som kommer från klienten kan liknas vid en stig som leder till att ett visst kodstycke på servern exekveras.

URI	Metod	Beskrivning
Länk till API:t	Typ av API-anrop	Förklaring av API-anropet

Tabell 2: Exempel på hur API-anrop kommer implementeras på servern.

För att minska belastningen på klientens processorkapacitet strukturerades applikationen genom att all databearbetning sker på serversidan. På så vis kan arbetsbelastningen och laddningstiden reduceras för användaren vilket resulterar i en snabb och responsiv applikation.

4.2.2 Inloggningssystem

För att kunna autentisera att ett API-anrop kommer från en behörig användare krävs det att servern har kunskapen kring vilken användare som har rätt till vilken data. I databasen lagras information om användarens inloggningsuppgifter vilket möjliggör att servern kan verifiera varje API-anrop från användaren. Följaktligen skickas känslig data endast till behöriga användare.

Då lösenordet lagras i databasen krävs det specifika säkerhetsåtgärder utifall någon obehörig skulle få tillgång till databasen. Modulpaketet *hashlib* implementerades, vilket innehåller ett flertal hashfunktioner för Python. Dessa funktioner krypterar användarens lösenord innan de sparas i databasen, vilket gör att vid ett obehörigt intrång eller vid stöld av databasen skulle lösenorden vara obrukbara, se figur 11. För detta projektet användes hash-funktionen *pbkdf2-sha256* [27].

```
1 {  
2     "name": "John Doe",  
3     "email": "john@doe.com",  
4     "password": "$pbkdf2-sha256$291120$Wh"  
5 }
```

Figur 11: Exempel på en JSON-fil som sparar inloggningsuppgifter med ett krypterat lösenord i databasen.

4.2.3 Dataanalys och maskininlärning i applikationen

För att presentera förutsägelsen av morgondagens HRV körs dataanalysen, som beskrivs i avsnitt 5.5, i servern. Varje gång en ny datapunkt från användaren registeras i databasen uppdateras den individuella maskininlärningsmodellen och således kan den tränas på ett större dataset och ge mer träffsäkra förutsägelser. När klienten efterfrågar den individuella HRV-förutsägelsen returnerar servern detta via sitt API.

4.2.4 Inhämtning av data via externt API

Som tidigare beskrivits i avsnitt 4.1 ville applikationen kunna presentera historisk träningsdata på ett intuitivt sätt till användaren. Då databasen innehöll få träningspass behövdes fler importeras externt. Tränings tjänsten Strava har över 73 miljoner användare och används av flera i projektgruppen [28]. Syftet med Strava är att kunna lagra historiska träningspass och dela dessa med andra Strava-användare. Strava tillhandahåller ett öppet API, motsvarande det API som utvecklats i servern, vilket möjliggör att historisk individuell träningsdata kan hämtas från Strava och sedan lagras i databasen. Detta API implementerades i servern och en användare kan således få sin historiska träningsdata från Strava presenterad i mobilapplikationen.

5 Metod för studie

Målet med projektet är att ta reda på om de går att hitta samband från olika värden för att kunna optimera individens prestationsutveckling, för att ta reda på detta valde vi att göra en studie. Denna studie är tänkt att bygga vidare på en tidigare studie som endast använde Banister-modellen för att undersöka

om HRV-responser, som en representativ markör för träningsadaption, kan förutsägas utifrån elite rugby idrottares träningsbelastningar [29]. Syftet med denna studien är att undersöka om neurala nätverk som ANN och RNN kan förutsäga HRV-responser bättre än Banister-modellen genom att inkludera fler variabler. Det genomfördes genom att försökspersonerna under 40 dagar rapporterade in olika variabler i form av dagliga mätvärden. De primära variablerna var träningsintensitet och träningens varaktighet för att beräkna träningsbelastning. Då HRV påverkas av andra externa faktorer, se avsnitt 2.1, inkluderades även variablerna sömntimmar, stressnivå, humör, muskelvärk, energinivå och skador i ANN och RNN modellerna för att förutsäga HRV-responser.

5.1 Urval av försökspersoner

För denna studie rekryterades sex stycken försökspersoner, fem män och en kvinna i ålder 23.2 ± 1.6 år, längd 177.5 ± 3.9 cm samt vikt 71.3 ± 7.0 kg, där värdet efter \pm är standardavvikelsen. Fyra av försökspersonerna är författare till denna rapport och de två resterande är närstående till en av författarna. Fem av deltagarna är sedan tidigare aktiva idrottare medan en utav dem tränar sporadiskt. Deltagarna tränade badminton ($n = 1$), släggkastning ($n = 1$), löpning ($n = 2$), gym ($n = 3$), basket ($n = 2$) och boxning, ($n = 2$), där vissa även tävlade aktivt. Kravet för att delta i studien var att försökspersonerna skulle genomföra någon form av fysisk aktivitet under 40 dagar med minst fyra träningspass i veckan. Inget krav på vilken typ av fysiskt aktivitet sattes så länge aktiviteten gav någon grad av träningsstimulans. Ett krav som formulerades var dock att försökspersonerna skulle variera intensiteten på sina pass minst två gånger i veckan för att träningsstimulansen inte skulle vara likadan vid varje pass under de 40 dagarna. Huruvida detta uppnåddes genom att förlänga eller förkorta träningspasset, höja intensiteten eller sätta in ytterligare ett pass var upp till försökspersonerna själva.

5.2 Mätning av HRV

Försökspersonerna instruerades att utföra en mätning av HRV under 60 sekunder varje morgon direkt när de hade vaknat. De skulle avstå från att titta på mobilen, äta, dricka eller röra sig hastigt för att eliminera så många externa faktorer som eventuellt kunde påverka mätningen av HRV. Mätningen skulle genomföras oavsett om de hade tänkt träna den dagen eller inte. För att utföra mätningarna användes antingen Polar H7 eller Polar H10 Bluetooth pulsband¹ kopplat till den fritt tillgängliga mobilapplikationen². Försökspersonerna fick själva välja vilken position de utförde mätningen i, där alternativen var liggande, sittande eller stående. Ett krav var dock att försökspersonerna skulle fortsätta utföra mätningarna i samma position under hela studiens gång. Mätningen utfördes genom att sätta pulsbandet runt bröstet och följa instruktionerna i Elite HRV för utföra en HRV-mätning. Under denna mätning skulle försökspersonerna vara så avslappnade som möjligt och andas naturligt. Efter mätningen skulle de svara på en enkät, se första enkäten i appendix C, där de fyllde i HRV-värdet, antalet sömntimmar, stressnivå, skador, humörnivå, muskeltrötthet och energinivå. Alla parametrar, förutom HRV-värdet och sömntimmar, graderades mellan 0-10.

Efter att studien pågått i en vecka beslutades att ändra mättiden från en minut till tre minuter. Detta berodde på att den tidigare mättiden inte gav tillräckligt med tid för HRV-värdet att stabilisera sig och därmed ibland gav avvikande värden.

Efter att mätningen är klar utför Elite HRV några uträkningar, exempelvis Root mean square of the successive differences (RMSSD), se ekvation (14).

$$\text{RMSSD} = \sqrt{\frac{1}{N-1} \left(\sum_{i=1}^{N-1} ((R-R)_{i+1} - (R-R)_i)^2 \right)} \quad (14)$$

RMSSD ska reflektera variansen i hjärtfrekvensen och användes som en primär mätform för att uppskatta förändringar i HRV. Det har alltså påvisats vara den mest pålitliga mätformen för HRV [30]. Utöver att använda RMSSD valdes även att log-transformera RMSSD data med naturliga logaritmen då det reducerar variabilitet i data och därmed gör det enklare att tolka värdena [31].

¹En produkt av Polar Electro, Kempele, Finland

²En applikation av Elite HRV, Ashville, North Carolina, USA

En uträkning av det exponentiella glidande medelvärdet (EMA) av $\ln \text{RMSSD}$ utfördes, se ekvation (15).

$$\text{EMA} = \ln \text{RMSSD}(t) \cdot k + \text{EMA}(t-1) \cdot (1-k) \quad (15)$$

där t är idag, $t-1$ är gårdagen, N är tidsspannet och $k = 2/(N+1)$.

Den fyrtonde exponentiella glidande medelvärde användes. För $N = 40$ ger $\ln \text{RMSSD}_{40-\text{exp}}$ vilket är måttet som användes i kommande analyser (se avsnitt 5.5) som en representativ markör för träningsadaptation [29]. Vi kommer använda $\ln \text{RMSSD}$ istället för $\ln \text{RMSSD}_{40-\text{exp}}$ för att underlätta läsbarheten.

5.3 Mätning av träningsbelastning

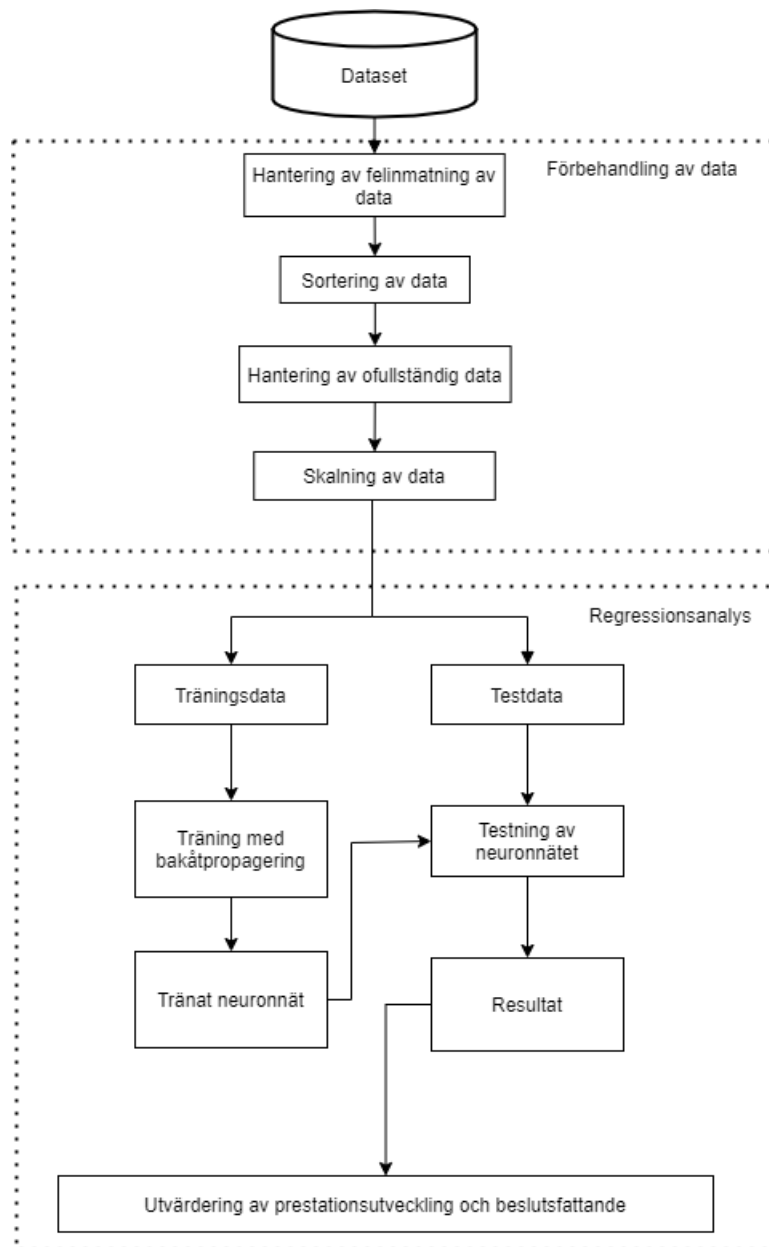
De träningsbelastningar som försökspersonerna genomförde mättes med hjälp av en modifierad sRPE-metod med en skala på 0-10, se tabell 3, för att underlätta graderingen för försökspersonerna. Försökspersonerna svarade på en enkät efter varje träningspass där de fyllde i träningsintensiteten som graderades på en skala 0-10, samt träningstyp, varaktighet och energinivån efter passet. Detta skulle helst utföras 30 min efter varje pass. Om försökspersonerna tränade flera gånger om dagen behövde de skicka in separata enkäter för varje träningspass. Det krävdes även att de skickade in en enkät på vilodagar där man endast graderade 0 på träningsintensiten. Försökspersonerna fick även instruktioner att använda sig av ett tidtagarur för att säkerställa den exakta längden på passet.

Gradering	Ansträngning
0	Vila
1	Mycket lätt
2-3	Lätt
4-5	Ganska ansträngande
6-7	Ansträngande
8-9	Mycket ansträngande
10	Maximalt ansträngande

Tabell 3: Tabell av Borgs modifierade RPE-skala.

5.4 Databearbetning

I denna studie studerades insamlad data för att hitta samband mellan olika parametrar och HRV. För att säkerställa att den insamlade datan är kompatibel för modellerna samt analys så behövde datan bli bearbetad. Databearbetning är en essentiell del för att kunna se tillförlitliga mönster som inte beror på exempelvis slumpen. Det kan vara värden som inte är inom det tillåtna intervallet, exempelvis varaktigheten i negativt antal minuter eller orimliga datakombinationer om värdet är ifyllt i fel spalt, samt saknade värden. Att analysera data utan att ha proaktivt bearbetat datan kan leda till att man får missvisande resultat. Med andra ord är det ytterst viktigt att datakvaliteten är hög, speciellt vid maskinlärningsprojekt där stora mängder data används för att träna modellen. Databearbetning är uppdelat i fyra steg, se figur 12.



Figur 12: Systemarkitektur över databearbetning samt regressionsanalys för ANN och RNN.

För att kunna bearbeta den insamlade data användes huvudsakligen biblioteken Pandas och NumPy. Pandas användes för att enkelt kunna läsa in och hantera datamängder medan NumPy användes för att hantera arrayer och enkelt kunna utföra beräkningar på de.

Insamlingen av data skedde genom enkäter där data lagrades i två separata datafiler, en som fylldes i efter träningspassen och en för uppmätt HRV, se appendix C.

5.4.1 Felinmatning av data

Felinmatning av data kunde endast förekomma i tre utav frågorna. Det berodde på att de flesta frågorna i formulären använde skalor från 0-10, se appendix C. Det som försökspersonerna behövde skriva in själva var sitt namn, typ av träning och varaktigheten av träningspasset. För att hantera felinmatning av namn genomfördes korrigeringar manuellt.

När det kommer till inmatning av varaktigheten på passen instruerades försökspersonerna att mata in det i antal minuter. Om de tränade i 100 minuter skulle de fylla in "100" i formuläret. Däremot förekom exempelvis att försökspersonerna matade in "100 min", "100 minuter" eller "1:40". Initialt korrigerades

det manuellt men valdes sedan att lösas genom att skriva en funktion som undersökte datamängden och konverterade datan till den önskade typen.

5.4.2 Sortering av data

Data för alla försökspersoner samlades in i en tabell. För att analysera respektive försöksperson krävdes det att datan sorterades in i separata tvådimensionella datastrukturer. En av dessa innehöll värdena för HRV, stress, humör med flera samt en för träningsintensitet och varaktighet av träningspass. Det vi undersökte i studien var om träningsbelastningen, stress, humör med flera påverkade morgondagens HRV-mätning. Datumen som automatiskt registreras när en enkät skickas in användes för att matcha HRV-värdet till den korresponderade träningsbelastningen. Det förekom att försökspersoner glömde skicka in antingen enkäten för träningspasset eller för mätning av HRV. Detta hanterades genom att ha med frågan "Träningsdag" i enkäten "Hur var din träning". Därmed kunde försökspersonerna skicka in enkäten för ett träningspass dagar efter genom att välja vilken dag träningspasset genomfördes. För HRV mätningen krävdes det däremot att det skickas in samma dag.

5.4.3 Ofullständig data

Om försökspersonerna hade skickat in en enkät utan att svara på alla frågor tilldelades ett null-värde för den information som saknades. För att hantera de saknade värdena användes metoden medelvärdesimputering. Det innebär att man ersätter det saknade värdet hos en variabel med medelvärdet av de observerade värdena för samma variabel [32]. Ett annat alternativ var att ta bort raderna med null-värden helt men på grund av de begränsade tidsspannet av studien så användes medelvärdesimputering.

5.4.4 Skalning av data

För modellerna ANN och RNN är det viktigt att den data som de tränas och testas på inte har för stor variation då det kan leda till lärningsprocessen i neuronerna går långsamt eller har svårt att lära sig [33]. Dessutom ser skalningen till att variabler med olika enheter bidrar med lika mycket till regressionsanalysen. Exempelvis kan träningsbelastningen anta värden upp till 1000 om en försöksperson graderar träningsintensiteten till 10 och träningen varar i 100 min medan värden för energinivå, stress, humör etc. antar värden mellan 0-10. I vår analys väger alla variabler lika mycket och därmed valdes det att skala alla indata variabler med Min-Max skalning [34]. Det valdes specifikt att skala till intervallet $[0, 1]$ då variablerna endast antar icke-negativa värden. Det gjordes genom att utgå från formeln i ekvation (16).

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (16)$$

där x är det originala värdet, x' är det skalade värdet, $\min(x)$ och $\max(x)$ är det minsta respektive största värdet som x kan anta.

5.5 Regressionsanalys

För alla tre modeller valdes delningsförhållandet 75% till träningsdata och 25% till testdata. Med 40 datapunkter per försöksperson blev det 30 datapunkter för att träna upp modellen och sedan 10 datapunkter för att utföra förutsägelserna. För studien som tidigare hade modellerat träningsbelastning och HRV användes delningsförhållandet 50% där 28 datapunkter användes till träningsdata och 28 datapunkter användes till testdata [29]. Eftersom vi hade färre datapunkter ansåg vi att 20 datapunkter var för lite och därmed tränades modellen upp på 30 datapunkter. Figur 12 visar i andra boxen hur regressionsanalysen kommer gå till för neuronnäten.

5.5.1 Banister-modellen

Banister-modellen som beskrivs i avsnitt 2.3.2 modellerar relationen mellan träningsbelastning (indata) från pulsdata och prestationsutveckling (utdata) vid löpning. Denna studie har istället valt att modellera relationen mellan idrottarnas subjektiva gradering av träningsbelastning (indata) och HRV (utdata), där vi använde mätformen $\ln \text{RMSSD}$, utifrån samma Banister-modell [29]. Detta implementeras i Python genom att utgå från ekvationerna (3), (4) och (5).

Från ekvation (3) och (4) så sattes fitness och trötthetsfaktorn till att vara

$$g(t) = g(t - i)e^{-i/\tau_1} + s(t) \quad (17)$$

respektive

$$f(t) = f(t - i)e^{-i/\tau_2} + s(t) \quad (18)$$

där $s(t)$ är den upplevda träningsbelastningen som försökspersonen har genomgått vid tiden t , där tiden mäts i dagar. Trötthetsfaktorn valdes att beskrivas med $f(t)$ istället för $h(t)$ då $h(t)$ är tänkt att beskriva HRV i nedanstående del.

På samma sätt ändrades ekvation (5) till

$$h(t) = h_0 + k_1 g(t) - k_2 f(t) \quad (19)$$

där $h(t)$ är HRV vid tiden t med ett initialt värde h_0 på HRV. En omskrivning (see appendix D.1 för härledningen) ger att

$$h(t) = h_0 + k_1 \sum_{i=1}^{t-1} s(i)e^{-(t-i)/\tau_1} - k_2 \sum_{i=1}^{t-1} s(i)e^{-(t-i)/\tau_2} \quad (20)$$

Modellen karaktäriseras, som tidigare nämnts, av två förändringstermer (k_1 och k_2), två tidskonstanter (τ_1 och τ_2) och ett initialt HRV-värde h_0 . Dessa parametrar bestämdes genom att minimera medelkvadratfelet (MSE) mellan den estimerade och den uppmätta HRV. Det vill säga

$$\text{MSE} = \frac{1}{n} \sum_{t=1}^n (h_t - \hat{h}_t)^2 \quad (21)$$

minimerades där h_t är det uppmätta HRV och \hat{h}_t är den förutsagda HRV-värdet.

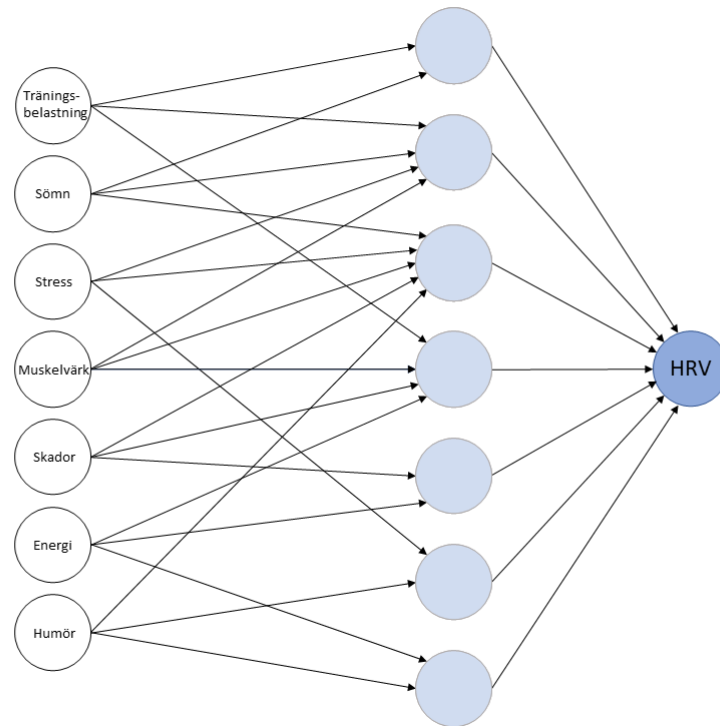
För att minimera ekvation (21) användes **optimize** från paketet SciPy som minimera objektiva funktioner genom att lägga in ett initial värde. En rutnätssökning användes på de första 30 dagarna för att erhålla värdena för parametrarna. De initiala värdena togs från studien som tidigare hade modellerat träningsbelastning och HRV [29]. Förändringstermerna k_1 och k_2 sattes till $7.67 \cdot 10^{-5}$ AU respektive $8.93 \cdot 10^{-5}$ AU där de inte ingick i rutnätssökningen då MSE inte påverkades av de. För τ_1 och τ_2 testades värden mellan 0-50 dagar för att sedan erhålla värdena i tabell 4 vilket gav det lägsta MSE. De estimerade parametrarna användes sedan för att förutsäga lnRMSSD-värdena de 10 efterföljande dagarna.

Försöksperson #	h_0	τ_1	τ_2
1	4.2	4	9
2	3.9	49	32
3	4.0	46	43
4	2.7	45	16
5	4.2	15	2
6	4.8	2	6

Tabell 4: Tabell över parametrarna för varje försöksperson till Banistermodellen

5.5.2 ANN

För ANN-modellen användes maskininlärningsalgoritmen Multi-Layer Perceptron (MLP). MLP är den mest grundläggande versionen av ett framåtpropagerande ANN och består av åtminstone tre lager av neuroner: ett inputlager, ett eller flera dolda lager och ett utdatalager. Figur 13 visar det neurala nätverket som tränades.



Figur 13: Figuren illustrerar det artificiella neuronnätet som användes för studien.

För att optimera algoritmen sattes parametrarna initialt till *solver* till '*lbfgs*', *activation* till '*relu*' och *max_iter* till 1500. Parametern *hidden_layer_sizes* itererades mellan 1 till 50 för varje enskild individ och värdet som minimerade respektive förlustfunktion valdes, se tabell 5.

Individ #	Antal dolda lager
1	44
2	8
3	49
4	47
5	15
6	4

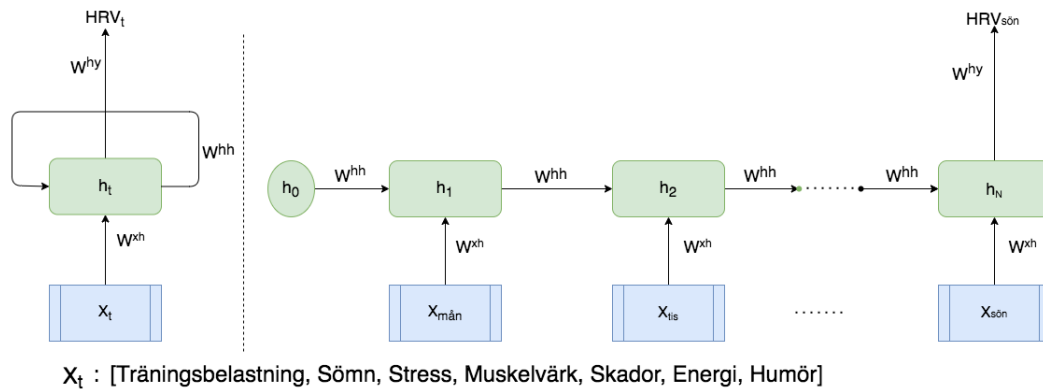
Tabell 5: Tabell över antalet dolda lager som minimerade förlustfunktionen för respektive individ.

5.5.3 RNN

RNN-modellen som implementerades beskrevs i avsnitt 2.3.3. Träningsdatan bearbetades enligt avsnitt 5.4 och delades upp i vektorer om sju dagar var. Anledningen till valet av sju dagar var då det existerande datasetet hade ett begränsat antal datapunkter från projektets studie. Således analyserar RNN-modellen indatan i en period av sju dagar för att förutsäga dagens HRV-värde. Detta illustreras i figur 14 där data från $X_{\text{mån}}$ till $X_{\text{sön}}$ förutsäger HRV-värdet $HRV_{\text{sön}}$. Respektive indata, X_t , bestod av en vektor med elementen [*Träningsbelastning*, *Sömn*, *Stress*, *Muskelvärk*, *Skador*, *Energi*, *Humör*].

Datasetet delades upp i tränings- respektive testdata där RNN-modellen kunde tränas och valideras. RNN-modellen är ett objekt från klassen *sequential* i Keras. Klassen *sequential* möjliggör att olika lager kan sammanfogas i en och samma RNN-modell. Den tränade modellen konstruerades med nio lager:

- 4 *LSTM*-lager med 100, 50, 20 respektive 10 neuroner vardera.
- 4 *dropout*-lager som slumpmässigt droppar 20% av enheterna i de första 3 *LSTM*-lagren.
- 1 *dense*-lager som förutsäger HRV värdet för den sista tidsenheten.



Figur 14: Diagrammet illustrerar RNN modellen som används för studien

Funktionaliteten som dropout-lagret bidrar med är att minska risken för att modellen överanpassar sig till träningsdatan, så kallad *overfitting*. Dense-lagret är ett lager där alla neuroner i lagret är kopplat till alla neuroner i det föregående lagret och används för att kunna förutsäga utdatan.

För att kalibrera RNN-modellen så itererade modellen över olika parametervärden. Optimeringen av RNN-modellen utvärderades genom att minimera felmarginalen samt att modellens förutsägelser följer den verkliga utdatan. Kalibreringen resulterade i att aktiveringsfunktionen sattes till 'huber loss' och antalet iterationer var optimalt vid 5-15 stycken.

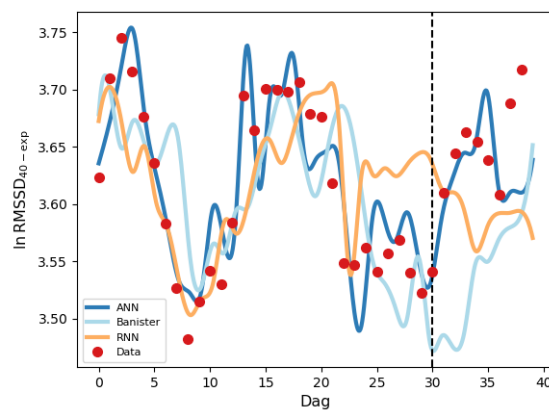
6 Resultat

Arbetet resulterade i en jämförelse mellan olika modeller för att förutse HRV i kombination med en mobilapplikation som bland annat kan visa användarens träningspass och förutspådda HRV. De två delarna kopplas ihop med en server som hämtar, beräknar och skickar data. Resultatet av projektet är en plattform som tillåter fortsatt vidareutveckling.

6.1 Jämförelse av modeller

Resultatet från de tre olika modellerna Banister, ANN och RNN visas i figurerna 15, 16 och 17. I respektive figur presenteras även den uppmätta lnRMSSD-datan för försöksperson #2, #4 och #5. De resterande försökspersoners resultat finns att se i Appendix D. Den svarta sträckade linjen i figurerna delar upp träningsdata från testdata. Mellan dag 0 och 30 har respektive modell tränats upp för att sedan förutsäga värden för de 10 resterande dagarna.

Försöksperson #2 är idrottaren som tränade sporadiskt innan studien. För #2 visas lnRMSSD-värden i figur 15. Uppmätta värden för lnRMSSD, som representeras av de röda cirkarna, stiger de första tre dagarna för att sedan avta fram till dag 7. Detta mönster ser ut att upprepa sig över de 40 dagarna. För ANN syns en tydlig överanpassning till träningsdatan medan Banister och RNN följer träningsdata relativt bra hela 40 dagarsperioden, se r -värden i tabell 7. När modellerna använder testdata för att förutsäga lnRMSSD ser vi att ANN presterar markant bättre än RNN och Banister-modellen. ANN lyckas fänga mönstret av lnRMSSD mellan dag 30-37 för att sedan underestimera de resterande punkterna. ANN presterar därmed bäst av de tre modellerna, se tabell 6 för en mer detaljerad jämförelse av MSPE.



Figur 15: Uppmätt \ln RMSSD för försöksperson #2 visas tillsammans de tre modellerna Banister, ANN och RNN.

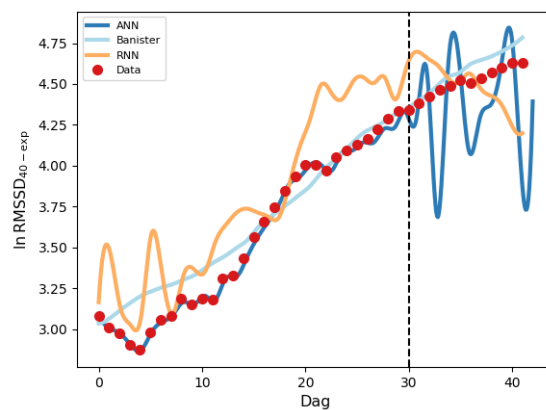
Försöksperson #	Banister	ANN	RNN
1	0.00061	0.00173	0.00030
2	0.00810	0.00150	0.00721
3	0.01080	0.06457	0.03569
4	0.00702	0.12867	0.05252
5	0.00478	0.00614	0.00155
6	0.00913	0.00138	0.00036

Tabell 6: Jämförelse av MSPE för Banister-, ANN- och RNN-modellerna för varje försöksperson.

Försöksperson #	Banister	ANN	RNN
1	-1.57	0.98	0.44
2	0.24	0.83	0.58
3	0.68	0.99	0.78
4	0.87	0.99	0.82
5	0.73	0.35	0.82
6	0.41	0.87	0.45

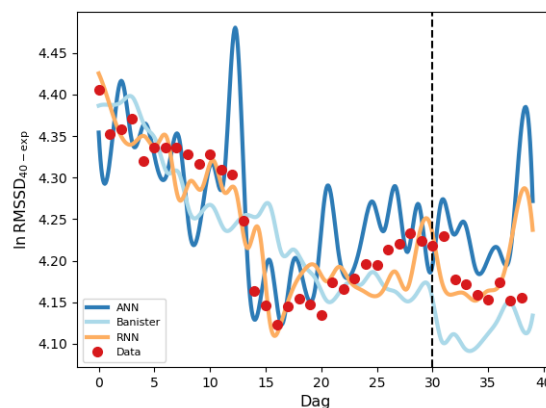
Tabell 7: Jämförelse av r-värde för Banister-, ANN- och RNN-modellerna för varje försöksperson.

Försöksperson #4 är en av idrottarna som tränade regelbundet innan studien. För #4 visas \ln RMSSD-värden i figur 16. De uppmätta \ln RMSSD minskar de första 5 dagarna för att sedan öka fram till dag 40. Jämfört med Figur 15, observerar vi att Banister-modellen för försöksperson #4 fångar mönstret i den uppmätta \ln RMSSD relativt bra för träningsdatan. Däremot ser vi att ANN överanpassas i hög grad till träningsdatan medan RNN-modellen tydligt överestimerar, se tabell 7. För testdata observerar vi att ANN:s förmåga att förutsäga \ln RMSSD är undermålig. För RNN förutspås att \ln RMSSD ska minska de 10 sista dagarna när de i själva verket stiger. Det trenden lyckas Banister fånga relativt bra med en minimal överestimation efter dag 35, se tabell 6 för en mer detaljerad jämförelse av MSPE.



Figur 16: Uppmätt \ln RMSSD för försöksperson #4 visas tillsammans de tre modellerna Banister, ANN och RNN.

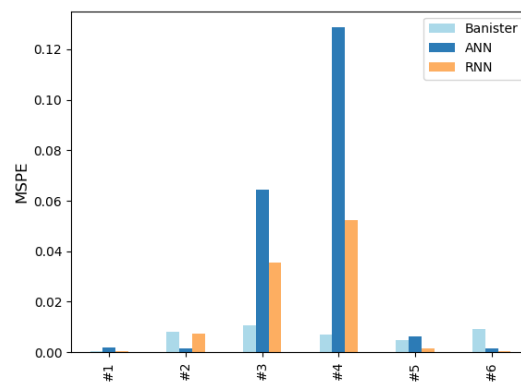
Försöksperson #5 är också en av idrottarna som tränade regelbundet innan studien. För #5 visas \ln RMSSD-värden i figur 17. Den uppmätta \ln RMSSD saknar ett tydligt mönster, dock observeras en tydlig minskning mellan dag 10 och 15 för att sedan ökas fram till dag 30. Endast RNN-modellen lyckas finna trenden på träningsdata, dock i form av överanpassning, se tabell 7. För testdata förutsäger RNN relativt bra de första 6 dagarna för att sedan överestimera resterande dagarna. Anpassningsgraden för ANN-modellen till träningsdata är avsevärt sämre jämfört med försöksperson #2 och #4 vilket får modellen att överestimera markant på testdata. Banister-modellen tränas heller inte tillräckligt vilket får den att underestimera på testdatan.



Figur 17: Figur över uppmätt \ln RMSSD för försöksperson #5 utritad tillsammans de tre modellerna Banister, ANN och RNN.

En jämförelse av MSPE för alla tre modellerna för varje försöksperson visas i Figur 18. För fyra av försökspersonerna #1, #3, #4 och #5 observeras att ANN är sämst på att förutsäga \ln RMSSD av de tre modellerna. För försöksperson #1, #5 och #6 observeras att alla modeller har lågt MSPE där RNN presterade bäst för dessa tre försökspersoner, se tabell 6.

Genom att ta medelvärdet av MSPE för alla sex försökspersoner observeras att av de tre modellerna har Banisters förutsägelser lägst felmarginal, RNN näst lägst och ANN högst, se tabell 8. För träningsdata uppvisade ANN- och RNN-modellerna höga korrelationer mellan modellerat och uppmätt \ln RMSSD_{exp} medan Banister-modellen hade relativt låga r -värden, se tabell 9.



Figur 18: Histogram över MSPE för alla tre modellerna Banister, ANN och RNN för varje försöksperson.

	Banister	ANN	RNN
MSPE	0.006 ± 0.020	0.034 ± 0.048	0.016 ± 0.020

Tabell 8: Jämförelse av med hjälp av standardavvikelse $\mu \pm \sigma$ av MSPE för varje modell.

	Banister	ANN	RNN
r	0.20	0.84	0.64

Tabell 9: Jämförelse mellan medelvärdet av r -värde för varje modell.

6.2 Server

Kommunikation mellan klient och server sker via API-anrop och de funktioner som implementerades visas i tabell 10. Datan som lagras i databasen i MongoDB hämtas och modifieras i servern för att sedan skickas till klienten. Servern byggdes som ett REST-API vilket möjliggör att flera klienter kan hanteras parallellt. Servern implementerades i Python med hjälp av ramverket Flask. Både API-anrop och serversvar strukturerades i JSON-format, vilket är ett format som både servern och klienten enkelt kan tolka. Validering av data från klienten sker på serversidan, vilket exempelvis säkerställer att samma träningsformulär inte kan laddas upp två gånger i databasen.

API för autentisering

URI	Metod	Beskrivning
/signup	POST	Registrera en ny användare i databasen.
/login	POST	Autentiserar användaren via matchning av uppgifter i databasen.

API för formulär

URI	Metod	Beskrivning
/api/form/stats	GET	Hämtar individens morgonformulär från databasen.
/api/form/stats	POST	Postar individens morgonformulär till databasen.
/api/form/stats	DELETE	Tar bort ett specifikt morgonformulär från databasen.
/api/form/stats/hrv	GET	Hämtar en individs specifika HRV-värde från databasen.
/api/form/stats/all/hrv	GET	Hämtar individs alla HRV-värden från databasen.
/api/form/training	GET	Hämtar individens träningsformulär från databasen.
/api/form/training	POST	Postar individens träningsformulär till databasen.

API för aktiviteter

URI	Metod	Beskrivning
/api/activities	GET	Hämtar en individs träningspass från databasen.
/api/activity	POST	Postar en individs träningspass till databasen.

API för grupper och organisationer

URI	Metod	Beskrivning
/api/group	GET	Hämtar alla medlemmar i en specifik grupp från databasen.
/api/user/group	POST	Skapar en ny grupp och adderar en individ till gruppen.
/api/user/group	GET	Hämtar alla grupper som en individ är medlem i.
/api/organisation	GET	Hämtar alla medlemmar i en specifik organisation från databasen.
/api/user/organisation	POST	Skapar en ny organisation och adderar en individ till organisationen.
/api/user/organisation	GET	Hämtar alla organisationer som en individ är medlem i.
/get_png	GET	Hämtar logon till gruppen/organisationen från databasen.

API för att hämta predikterat HRV

URI	Metod	Beskrivning
/ml/neural	GET	Hämtar morgondagens predikterade HRV-värde, baserat på ANN.

API för inhämtning av data från Strava

URI	Metod	Beskrivning
/strava/authorize	GET	Autentiserar användaren så att data kan hämtas från Strava.
/strava/connect	POST	Sammankopplar existerande träningsdata med data från Strava.
/strava/athlete	GET	Hämtar individens användaruppgifter i Strava från databasen.
/strava/athlete_all	GET	Hämtar alla individer som har lagrat data från Strava i databasen.

Tabell 10: Tabell över alla API-anrop som implementerades på servern.

6.3 Applikation

Majoriteten av alla planerade sidor blev implementerade i applikationen. En del funktionalitet saknas men visuellt sett är den huvudsakligen färdig. Servern och applikationen är ihopkopplade vilket bland annat möjliggör användarhantering och inloggning. Efter inloggning visas dashboarden som kan visa datum och temperatur i realtid utifrån användarens fysiska plats. Dashboarden innehåller information om morgondagens förväntade HRV och möjlighet att navigera till en ny sida där sRPE och HRV kan fyllas i. Denna data skickas sedan till servern som analyserar och sparar datan i databasen. Dessa värden finns sedan tillgängliga under historiksidan. Applikationen kan också låta användaren starta ett nytt träningspass som sparas i servern när det har avslutats och finns sedan tillgängligt under historiksidan. I dagsläget mäter träningspasset endast tid.

En del önskad funktionalitet blev inte färdigställd. Grupper bestående av olika användare kan skapas och visas men det finns ingen möjlighet att förändra vilka användare som ingår i en grupp. Däremot

kan ett flöde av medlemmarnas kombinerade aktiviteter visas i datumordning. Profilsidan visar hur olika personliga mål kan se ut och ska innehålla information om användaren, dock finns ingen koppling mellan denna sida och servern.

7 Diskussion

Det primära syftet med studien var att undersöka om ANN och RNN är bättre än Banister-modellen på att förutsäga $\ln RMSSD$ genom att inkludera parametrar som träningsbelastning, sömn, stress, muskelvärk och humör. Målet med att skapa en applikation var att skapa en plattform med möjlighet till intuitiv information utifrån analysering av användarens träningsdata. Då utveckling av applikationen och genomförandet av studien gjordes parallellt innebar detta att dessa delar inte alltid var kompatibla med varandra under arbetets gång. I följande underavsnitt kommer först studien diskuteras kring metodval, resultat och jämförelse med resultat från andra studier. Sedan diskuteras implementationen av server, databas och applikation följt av etiska aspekter.

7.1 Val av mätmetod

Som vi beskrev i avsnitt 2.2 finns det en rad olika sätt att mäta träningsbelastning för en idrottare, där träningsbelastning delas upp i extern och intern belastning.

TRIMP-metoden anses vara tillförlitlig då den använder ”objektiv data” och var en kandidat vid val av mätmetod, där planen var att mäta puls under olika träningspass. En nackdel med att mäta puls är dock att detta inte alltid reflekterar träningsbelastning där ett exempel är löpning gentemot styrketräning. Således lämpar sig TRIMP främst vid aktiviteter med kontinuerlig puls som till exempel cykling och löpning. Detta skulle ha resulterat i att antalet tillgängliga försökspersoner skulle ha begränsats från sex stycken till två stycken. sRPE är en annan metod som kan appliceras inom diverse idrottsaktiviteter och tillåter en större mängd idrotts- och sportaktiviteter, vilket skulle tillåta alla tillgängliga försökspersoner att delta i studien. En nackdel är att mätningar med sRPE har svårare att uppfatta förändringen i hjärtfrekvens för extremt intensiv träningsbelastning vilket ger en mätning som inte är tillförlitlig. En avvägning gjordes därför mellan antalet försökspersoner till studien och tillförlitligheten av mätningarna. Det är däremot något som är värt att studera vidare på hur olika mätmetoder som TRIMP eller en mätning av träningsbelastning som hastighet med hjälp av GPS kan användas för att förutsäga HRV-responser. Det som däremot är värt att nämna att sRPE metoden är både ett kostnadseffektivt val och tillräckligt tillförlitligt [15]. Om man har begränsade resurser eller tillgångar kan sRPE vara ett utmärkt val.

7.2 Val av värde att förutsäga

Då modellering av träning, träningsprogram och förhållandet mellan träning och vila är ett område med stor utvecklingspotential är det litterära underlaget begränsat. Den mest populära modellen att använda för detta är Banister-modellen, som finns beskriven i 2.3.1. Ett problem med att använda prestationsutveckling som värde att förutsäga är att det skulle ha krävt ett flertal träningspass med maximal ansträngning [17]. Då flera av försökspersonerna inte hade möjlighet att göra stora förändringar i sina träningsplaner var följdaktligen prestationsutveckling inte lämpligt som värde att förutsäga. Det slutliga valet blev att använda HRV, vilket inte krävde någon typ av förändring på befintliga träningsplaner.

7.3 Tidsspann och datamängd

Studien genomfördes med 6 försökspersoner och pågick under 40 dagar. En datapunkt per person och dag samlades in oavsett om de tränade eller inte. De begränsade antalet datapunkter påverkade storleken på träningsdatan, vilket vanligtvis är en hämmande faktor för inlärning av en modell. För att matematiskt modellera träningsbelastning och prestationsutveckling argumenterar Morton för att 60-90 datapunkter är att föredra [17].

För Banister-modellen observerades i studien en anpassningsgrad på $r = 0.2$. För de individuella försökspersonerna varierade r -värdena mellan -1.57 till 0.8735 . Det observerade värdet från projektets studie var lägre än värdet som Williams et al. (2018) uppmätte till $r = 0.66$ när elitrugbyspelare studerades. Banister-modellen antar ett lägre medelvärde på anpassningsgraden på grund av det låga

r -värdet ($r = -1.57$) för försöksperson #1. För de andra försökspersonerna erhöles relativt höga korrelationer mellan den modellerade och uppmätta \ln RMSSD i enlighet med studien av Williams et al. (2018). Fördelen med en maskininlärningsmodell jämfört med Banister-modellen är att den kan ta hänsyn till fler parametrar.

ANN-modellen och de uppmätta \ln RMSSD hade en hög anpassningsgrad på $r = 0.84$. Detta indikerar att modellen är överanpassad till träningsdatan. Överanpassning kan leda till en försämrad förutsäggelseförmåga, vilket kan vara en förklaring till varför Banister får ett lägre medelvärde för MSE än ANN trots att ANN använder fler parametrar. Det som däremot är positivt med ANN-modellen jämfört med Banister är att desto mer data som används desto bättre kan den tränas för att inte överanpassa och därmed kunna göra bättre förutsägelser på ny indata [35].

Trots en begränsad mängd av träningsdata så presterade RNN-modellen relativt bra jämfört med Banister- och ANN-modellen. Detta kan förklaras av att RNN-modellen i studien använde flera observationer i en tidsekvens av 7 dagar för att förutsäga HRV. HRV påverkas, vilket nämns i avsnitt 2.1, av både livsstils- och miljöfaktorer. Till exempel kan den fysiologiska effekten av muskelsmär och energinivå från träningspassen ackumuleras över tid där inte bara gårdagens muskelsmär, energinivå och träningsbelastning påverkar dagens HRV utan även tidigare dagar kan påverka [36]. Därav anses RNN, som tar hänsyn till tiden, vara ett lämpligt val för att undersöka hur de olika parametrar som träningsbelastning, muskelsmär, energinivå etcetera kan påverka HRV. Med fler datapunkter går det också att mata in fler tidssekvenser och använda ett längre tidsspänn för att eventuellt få ett bättre resultat.

Även om Banister gav lägst MSPE så är $r = 0.2$ för lågt för att kunna fånga komplexiteten i HRV-responser vilket medför underanpassning till träningsdatan och därmed troligen dålig förutsäggelseförmåga av nya HRV-responser [37]. Däremot så har RNN ett relativt lågt MSPE med tillräckligt högt r -värde för att med tillräckligt noggrannhet förutsäga HRV-responser. Det vill säga för försökspersonerna med positiva r -värden och låga MSE kan en upptränad RNN-modell förslagsvis användas som alternativ för att förutsäga framtida HRV-värden.

7.3.1 Urval av variabler i analysen

Att välja vilka variabler man vill inkludera i analysen är kritiskt för resultatet. Det kan förekomma att vissa variabler i modellen resulterar i bias eller så kan värdena bidra till att modellen presterar bättre men detta kan vara av slump. På grund av tidsbrist samt att två undersökningar pågick samtidigt i denna studien så valdes att inkludera alla variabler som undersöktes i studien för nerronnät modellerna. I framtida arbeten kan en välja att analysera vilka variabler som bidrar till en försämrad förutsäggelseförmåga av modellerna för att exkludera de från analysen.

7.3.2 Framtida studier

Syftet med detta projekt från början var att analysera den insamlade datan och hitta ett sätt att optimera idrottarens träning baserad på några parametrar. Datainsamlingsprocessen tog lång tid och det ledde till att en del undersökningar som var planerade inte blev av. En tanke var att se om man kan påverka en individs HRV genom att justera parametrar såsom sömn och träningsbelastning. Nästa dags HRV skulle kunna förutsägas genom att mata in olika värden för sömn och träningsbelastning i modellen och sedan välja ut de värden som leder till högst HRV. Vi kunde inte genomföra en sådan studie på grund av tidsbrist samt att den mängd data som samlades inte var tillräcklig för att göra tillförlitliga förutsägelser. Detta kan vara värt att studera i en framtida studie där man har mer data samt tid för att validera resultatet.

Ett annat intressant område att undersöka vidare kring hur modellerna kan simulera effekterna av olika träningsprogram på HRV för att objektivt kunna planera hur en atlet som rehabiliterar från en skada bör sätta upp sin träning [38].

7.4 Server och databas

Målet med servern och databasen uppfylldes under projektets gång. Även om slutresultatet blev önskvärt hade en tydligare planering av databasens struktur effektiviserat arbetet. Nya krav och funktioner upptäcktes under utvecklingens gång vilket medförde att databasen behövdes struktureras om. Exem-

pelvis krävde inhämtningen av data från Strava, som beskrivs i avsnitt 4.2.4, att aktivitetsdatan i den befintliga databasen strukturerades om för att matcha formatet.

Istället för ramverket Flask kunde ramverket Django ha använts för att bygga serverarkitekturen. Båda ramverken bygger på programmeringspråket Python. Fördelen med Django är att ramverket är mer komplett för större applikationer, dock kräver det mer kunskap och initial konfiguration. Om målet med projektet varit en storskalig plattform med hög kapacitet skulle Django således varit ett bättre alternativ. Valet föll dock på Flask som kunde driftsättas snabbare. Det faktum att serverarkitekturen inte behövde ändras under utvecklingens gång visar på att den valda arkitekturen var korrekt från start.

Implementationen av servern tog längre tid än planerat som en följd av gruppens begränsade erfarenheter inom API-utveckling samt programmering med Flask. Följaktligen gick en stor andel av utvecklingstiden åt till att inhämta kunskap. Gruppen är enig kring att implementationsfasen inte kunde utförts snabbare då kunskap kring andra serverlösningar saknades vid start. Förbättringspotentialen ligger i planeringsfasen, där projektplanen borde tagit hänsyn till inhämtningen av kunskap.

Testningen av serverns funktionaliteten var en process som borde planerats redan vid starten av projektet. En mer genomtänkt och automatiserad testningsprocess hade minskat antalet timmar som lades på felsökning. Exempelvis kunde testfunktioner som testar flera olika typer av indata direkt ha implementerats för att säkerställa att servern alltid fungerar som planerat.

Då känslig data hanteras i servern hade ytterligare säkerhetslager kunnat implementeras. I nuvarande version autentiseras användaren genom ett API-anrop till inloggningssystemet, se avsnitt 4.2.2. Genom att implementera ett ramverk som genererar slumpmässiga nycklar vid varje API-anrop, skulle servern ha möjlighet att kontrollera exakt vilka användare som har tillåtelse att använda olika API-funktioner. Detta har dock inte implementerats i nuvarande version, då en bred lansering av applikationen inte är inplanerad.

7.5 Applikationens funktionalitet och vidareutveckling

I appendix A finns bilder på applikationens alla sidor, där projektet lyckats få med de önskade funktionerna designmässigt. Däremot på grund av tidsbrist och att projektets alla delar inte varit helt synkroniserade så har inte alla önskade delar av applikationen implementerats än. Detta eftersom det fanns en ovisshet om vad vår studie skulle ge för resultat och om det skulle vara möjligt att förutspå morgondagens HRV. Ifall detta hade varit möjligt och en enkel, samt lättförståelig förklaring till HRV-värden hade getts så hade projektet uppnått sitt syfte att skapa en MVP. Ett annat sätt som applikationen hade kunnat nå, enligt projektets mening, en MVP skulle varit om till exempel ett pulsband hade kunnat kopplas till applikationen och användaren hade kunnat mäta sin puls under ett träningspass. Anledningen till att en MVP inte uppnåddes berode på tidsbrist och att vissa delar av projektet var svårare och mer tidskrävande än förväntat.

Utöver att skapa en MVP så var projektets syfte att ge applikationens användare lättförståelig information som skulle ge användaren återkoppling kring hur denne ska strukturera träningens frekvens och intensitet för optimal utveckling. Detta delmål har fått en bra start, om vi skulle göra fler och längre studier om olika värden som kan påverka träningen skulle vi kunna presentera dessa i appen och ge användaren verktyg för att planera sin träning på bästa sätt. Ytterligare en funktion som uppkom under projektet gång var att kunna lägga in egna mål, till exempel 10 km per vecka eller 10 000 steg varje dag. Dessa mål skulle användare kunna ha, samt gemensamma mål tillsammans med användarens vänner i en grupp. Detta var ett av flera exempel på funktioner som uppkom under projektet gång som inte var med i första planeringsfasen.

Att skapa just en mobilapplikation istället för ett datorprogram eller webbsida har sin grund i mobiltelefonens portabilitet och att stora delar av befolkningen redan använder den som hjälpmedel i andra vardagliga syften.

GDPR-processen för att få tillåtelse att lagra persondata i applikationen sker i den nuvarande versionen manuellt. I en framtida version skulle en sekretesspolicy kunna implementeras direkt i appen, där nya användare kan ge sin tillåtelse att persondata lagras i applikationen. Denna sekretesspolicy skulle då innehålla vad för slags information som samlas in, hur denna information samlas in och hur denna information lagras och skyddas. Fördelarna med att implementera en sekretesspolicy som kan accepteras direkt i applikationerna blir främst att friktionen för nya användare minskas och att det manuella pappersarbetet försvinner.

7.6 Samhälleliga och etiska aspekter

Att bevaka värden, exempelvis laktat och $VO_2\text{max}$, kan avslöja en hel del om formen på en idrottare och kan därmed bli en fråga om integritet. Detta gäller särskilt inom organiserad idrottsverksamhet där målet ofta är att producera elitidrottare. En risk med detta är att idrottare kan prioriteras bort om de inte uppvisar tillräckliga nivåer på specifika värden. Denna applikation skulle därmed kunna användas för att systematiskt diskriminera idrottare som inte uppvisar rätt värden på de faktorer som vi idag anser leder till en lyckad elitidrottare. Dock kan det vid elitnivå ibland vara befogat att mäta vissa värden för att säkerställa idrottarens hälsa och att den upprätthåller sin fysiska form. Den stora problemställningen uppstår ifall detta tänk börjar appliceras på unga idrottare som fortfarande är i starten av sina idrottskarriärer. Applikationen bör därmed användas med försiktighet och huvudsakligen för att följa upp ett träningsprogram efterföljs snarare än att bedöma en individuell idrottares fysiska form. För att motarbeta felanvändning av applikationen bör det övervägas att begränsa funktionaliteten för bevakning av värden till organisationer som är kopplade till riksidsrottsförbundet.

Under maj 2018 trädde General Data Protection Regulation (GDPR) i kraft i EU, vilket förenklat säger att det är olagligt att samla in data utan användarens vetskap. Enligt GDPR krävs det att persondata som entydigt kan kopplas till en individ hanteras på korrekt sätt och där användaren ger samtycke till detta. Enligt vår uppfattning faller syftet med insamlingen av persondata under artikel 6.1.a). Användare som tillåtit lagring av sin persondata har i projektet skrivit på GDPR-blanketten som visas i appendix B. För att förenkla denna process för användarna i framtiden skulle momentet kunna digitaliseras, vilket diskuteras i avsnitt 7.5.

Referenser

- [1] Metamatrix. *Vad är en dashboard?*. <https://www.metamatrix.se/vad-ar-en-dashboard/>, Uppdaterad: 2020-11-06, Hämtad: 2021-04-11.
 - [2] Medium. *What is grid search?*. <https://medium.com/fintechexplained/what-is-grid-search-c01fe886ef0a>, Hämtad: 2021-04-12.
 - [3] Karolinska Institutet. *Biomarkörer*. <https://mesh.kib.ki.se/term/d015415?fbclid=iwar31pbiuljrllu1lr-3x3cnsayh773j9o0yuwo2bpyfx5h-oicr57z1kcy>, Hämtad: 2021-05-12.
 - [4] Pitre C. Bourdon, Marco Cardinale, Andrew Murray, Paul Gastin, Michael Kellmann, Matthew C. Varley, Tim J. Gabbett, Aaron J. Coutts, Darren J. Burgess, Warren Gregson, and N. Timothy Cable. Monitoring Athlete Training Loads: Consensus Statement. *International Journal of Sports Physiology and Performance*, 12(s2):S2–161–S2–170, Apr 2017.
 - [5] G. Kenttä and P. Hassmén. Overtraining and recovery. A conceptual model. *Sports Med.*, 26(1):1–16, Jul 1998.
 - [6] Borejda Xhyheri, Massimiliano Mazzolini, Carmine Pizzi, and Raffaele Bugiardini. Heart Rate Variability Today. *Prog. Cardiovasc. Dis.*, 55(3):321–331, Nov 2012.
 - [7] Joshua A. Waxenbaum, Vamsi Reddy, and Matthew Varacallo. Anatomy, Autonomic Nervous System. In *StatPearls [Internet]*. StatPearls Publishing, Aug 2020.
 - [8] Clint R. Bellenger, Joel T. Fuller, Rebecca L. Thomson, Kade Davison, Eileen Y. Robertson, and Jonathan D. Buckley. Monitoring Athletic Training Status Through Autonomic Heart Rate Regulation: A Systematic Review and Meta-Analysis. *Sports Med.*, 46(10):1461–1486, Okt 2016.
 - [9] Sigita Kamandulis, Antanas Juodsnukis, Jurate Stanislovaitiene, Ilona Judita Zuoziene, Andrius Bogdelis, Mantas Mickevicius, Nerijus Eimantas, Audrius Snieckus, Bjørn Harald Olstad, and Tomas Venckunas. Daily Resting Heart Rate Variability in Adolescent Swimmers during 11 Weeks of Training. *Int. J. Environ. Res. Public Health*, 17(6):2097., Mar 2020.
 - [10] What is HRV? - Healthy Heart Rate Variability & How To Track It, Mar 2021. [Online; hämtad 14. Apr. 2021].
 - [11] Jin-Guo Dong. The role of heart rate variability in sports physiology. *Exp. Ther. Med.*, 11(5):1531–1536, May 2016.
 - [12] Stefan Sammito and Irina Böckelmann. Factors influencing heart rate variability. *International Cardiovascular Forum Journal*, 6(0), Maj 2016.
 - [13] Anthony Turner, Chris Bishop, Geoff Marshall, and Paul Read. How to monitor training load and mode using sRPE. *Professional Strength and Conditioning Journal*, Jan 2015.
 - [14] Luiz Perandini, Thiago Siqueira-Pereira, Nilo Okuno, Lúcio Soares-Caldeira, and Fabio Nakamura. Use of session RPE to training load quantification and training intensity distribution in taekwondo athletes. *Science & Sports - SCI SPORT*, 27(4), Sep 2012.
 - [15] Joao Henrique Falk Neto, Ramires Alsamir Tibana, Nuno Manuel Frade de Sousa, Jonato Prestes, Fabricio Azevedo Voltarelli, and Michael D. Kennedy. Session Rating of Perceived Exertion Is a Superior Method to Monitor Internal Training Loads of Functional Fitness Training Sessions Performed at Different Intensities When Compared to Training Impulse. *Front. Physiol.*, 11, Aug 2020.
 - [16] C. Foster. Monitoring training in athletes with reference to overtraining syndrome. *Med. Sci. Sports Exerc.*, 30(7):1164–1168, Jul 1998.
 - [17] Richard Morton, J. Fitz-Clarke, and E. Banister. Modeling human performance in running. *Journal of applied physiology (Bethesda, Md. : 1985)*, 69(3):1171–7, Oct 1990.
 - [18] Neural Network Definition, Apr 2021. [Online; accessed 25. Apr. 2021].
 - [19] Neural Network Definition, Apr 2021. [Online; accessed 23. Apr. 2021].
-

-
- [20] Yann Lecun, Leon Bottou, Genevieve Orr, and Klaus-Robert Müller. Efficient BackProp. *Research-Gate*, Aug 2000.
- [21] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Comput.*, 31(7):1235–1270, Jul 2019.
- [22] Pranoy Radhakrishnan. *Introduction to Recurrent Neural Network - Towards Data Science*. Towards Data Science, Jun 2018.
- [23] Robin M. Schmidt. Recurrent Neural Networks (RNNs): A gentle Introduction and Overview. *arXiv*, Nov 2019.
- [24] CS 230 - Recurrent Neural Networks Cheatsheet, May 2021. [Online; accessed 14. May 2021].
- [25] Jesús Gonzalez and Wen Yu. Non-linear system modeling using LSTM neural networks. *IFAC-PapersOnLine*, 51(13):485–489, Jan 2018.
- [26] Correlation - Statistical Techniques, Rating Scales, Correlation Coefficients, and More - Creative Research Systems, Sep 2019. [Online; accessed 14. May 2021].
- [27] LLC Assurance Technologies. passlib.hash.pbkdf2.digest - Generic PBKDF2 Hashes, 2020.
- [28] Inc Strava. Strava releases 2020 Year In Sport Data Report, 2020.
- [29] Sean Williams, Stephen West, Dan Howells, Simon P. T. Kemp, Andrew A. Flatt, and Keith Stokes. Modelling the HRV Response to Training Loads in Elite Rugby Sevens Players. *J. Sports Sci. Med.*, 17(3):402, Sep 2018.
- [30] Fred Shaffer and J. P. Ginsberg. An Overview of Heart Rate Variability Metrics and Norms. *Front. Public Health*, 5, 2017.
- [31] Changyong Feng, Hongyue Wang, Naiji L. U., Tian Chen, Hua H. E., Ying L. U., and Xin M. T. U. Log-transformation and its implications for data analysis. *Shanghai Arch. Psychiatry*, 26(2):105, Apr 2014.
- [32] Axel Ericsson and Johannes Zetterberg. *Om imputationsmetoder i statistisk analys : En simuleringsstudie om bortfallshantering och påverkan i en regressionsanalys*. 2020.
- [33] How to use Data Scaling Improve Deep Learning Model Stability and Performance, Aug 2020. [Online; accessed 13. May 2021].
- [34] Dmitrii Borkin, Andrea Nemethova, German Michalconok, and Konstantin Maiorov. Impact of Data Normalization on Classification Model Accuracy. *Research Papers Faculty of Materials Science and Technology Slovak University of Technology*, 27(45):79–84, Sep 2019.
- [35] How to Avoid Overfitting in Deep Learning Neural Networks, Aug 2019. [Online; accessed 14. May 2021].
- [36] T. Kajaia, L. Maskhulia, K. Chelidze, V. Akhalkatsi, and Z. Kakhbrishvili. ASSESSMENT OF EFFECTS OF NON-FUNCTIONAL OVERREACHING AND OVERTRAINING ON RESPONSES OF SKELETAL MUSCLE AND CARDIAC BIOMARKERS FOR MONITORING OF OVERTRAINING SYNDROME IN ATHLETES. *Georgian Med. News*, 311:79–84, Feb 2021.
- [37] Erik Aronsson and Olle Crondahl. *Maskininlärning applicerat på data över biståndsinsatser : En studie i hur prediktiva modeller kan tillämpas för analys på Sida*. 2017.
- [38] David C. Clarke and Philip F. Skiba. Rationale and resources for teaching the mathematical modeling of athletic training and performance. *Adv. Physiol. Educ.*, 37(2):134–152, Jun 2013.
-

A Applikationen

A mobile app login screen with a light purple background. It features two white input fields labeled 'E-mail' and 'password'. Below these are two purple buttons: 'Logga in!' and 'Sign up'.

Inloggningssida

A mobile app dashboard with a light purple background. At the top, it says 'Din Dagsform' and 'Lugn träning' next to a user profile icon. Below is a weekly calendar for 'Vecka: 19' with days 'Mån' through 'Sön' and a status 'Idag'. A weather forecast section shows 'Prognos: 14 Maj 2021' and '14.6 C °', along with 'HRV imorgon: 67' and a recommendation 'Din träningsintensitet imorgon bör vara: Normal'. A large purple button says 'Daily measure!'. At the bottom is a navigation bar with icons for a group of people, a profile, a home, a clock, and a document.

Dashboard/huvudsida

A mobile app sign-up screen with a light purple background. It features three white input fields labeled 'Username', 'E-mail', and 'Password'. Below these is a purple button labeled 'Sign up!'.

Tom inloggningssida.

A mobile app sign-up screen with a light purple background, showing filled-in data. The input fields contain 'Kalle', 'kalle@hotmail.com', and a masked password '.....'. Below these is a purple button labeled 'Sign up!'.

Ifylld inloggningssida.

Daglig HRV-mätning
Lugn träning

HRV: 68

Sömn: 7

Stressnivå: 4

Humör: 8

Muskeltrötthet: 6

Skador: 2

Förmulär för inmatning av HRV-mätning.

Lugn träning

HRV: 68

Sömn: 7

Stressnivå: 4

Humör: 8

Muskeltrötthet: 6

Skador: 2

Energinivå: 9

SPARA!

Förmulär för inmatning av HRV-mätning.

HRV - 8 maj 2021

Toppen!
Du har fyllt i dagens formulär.

För varje dag du fyller i formuläret kommer HRV prognosen förbättras.

Landningssida för lyckad inmatning av HRV-mätning.

Din Historik

flöde översikt värden

Datum	HRV	nåt annat
2021-05-14	68	3
2021-05-11	68	3
2021-05-08	120	3
2021-05-07	125	3
2021-04-29		

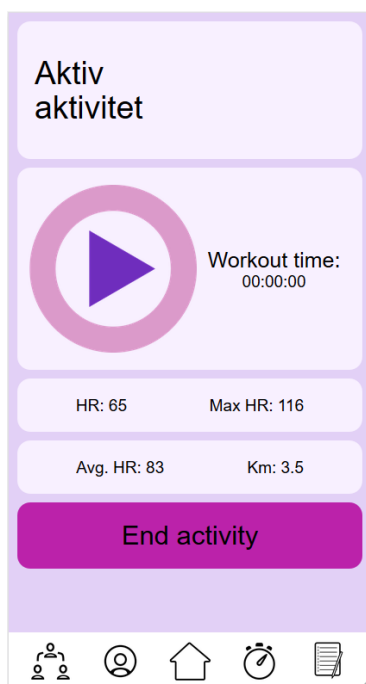
Historiksida som visar tidigare HRV-mätningar.



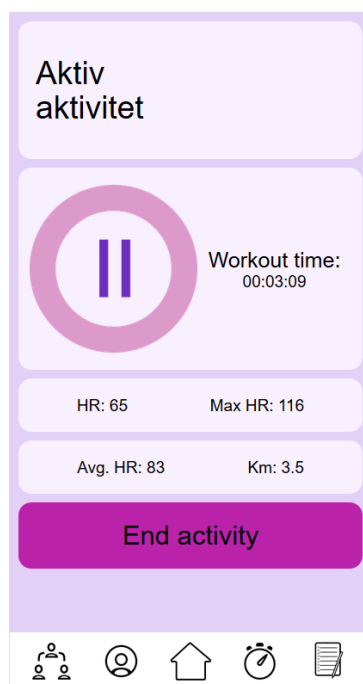
Tom sida för att skapa en ny aktivitet.



Ifylld sida för att skapa en ny aktivitet.



Sida för pågående aktivitet med pausad timer.



Sida för pågående aktivitet med aktiv timer.



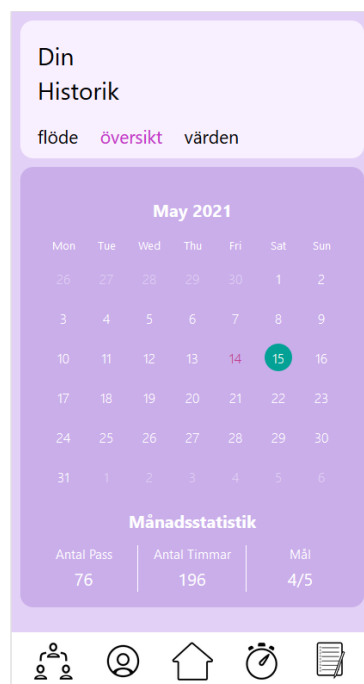
Historiksida som visar en användare med flera tidigare aktiviteter.



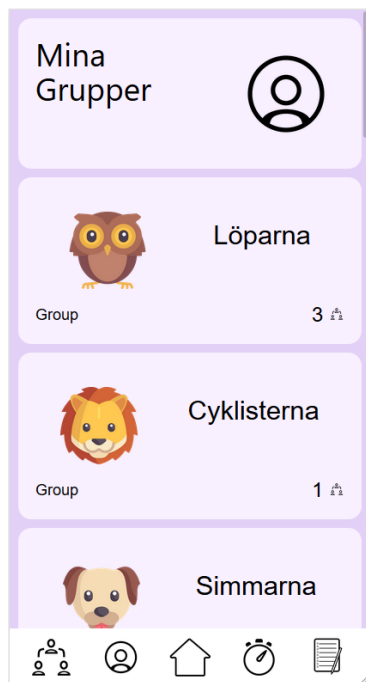
Historiksida som visar en användare med en tidigare aktivitet.



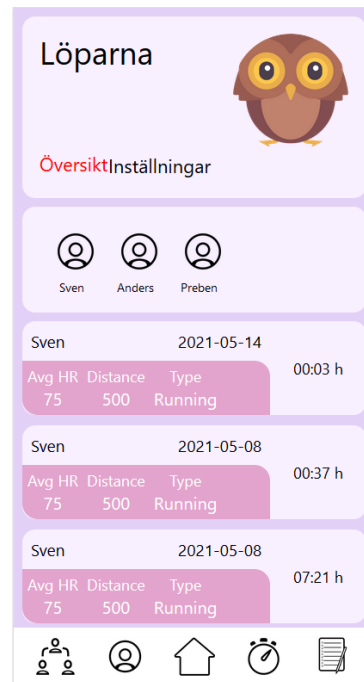
En detaljerad vy av en specifik tidigare aktivitet.



En kalendersida som ska visa när aktiviteter genomfördes.



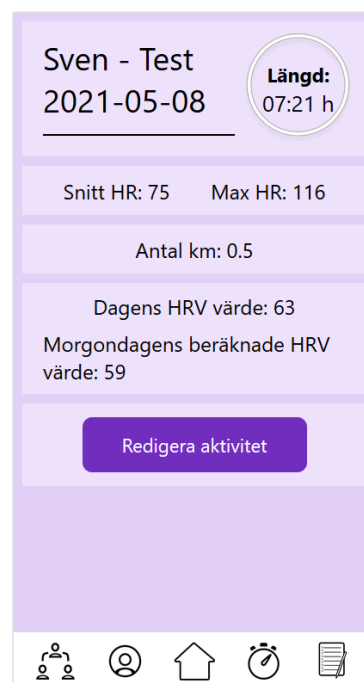
Översiktsida över grupper som användaren är medlem i.



Detaljerad sida för en specifik grupp.



Detaljerad sida för en specifik grupp med fokus på medlemmar-nas aktiviteter.



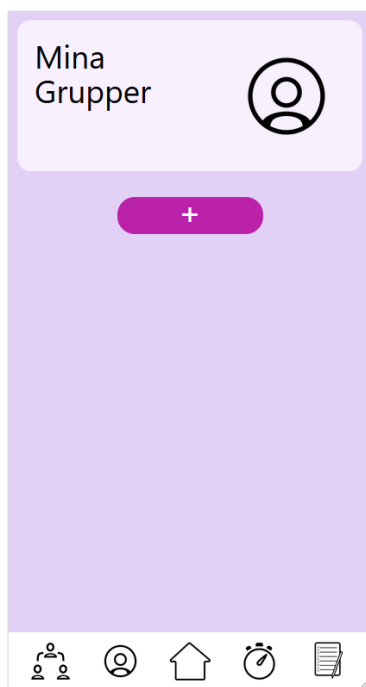
Detaljerad sida för en aktivitet för någon i en grupp.



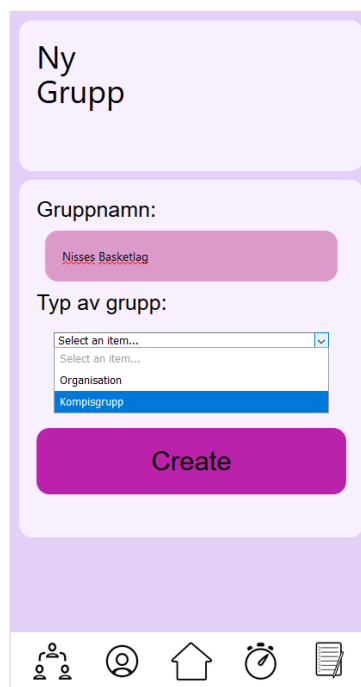
Sidan för inställningar för en specifik grupp.



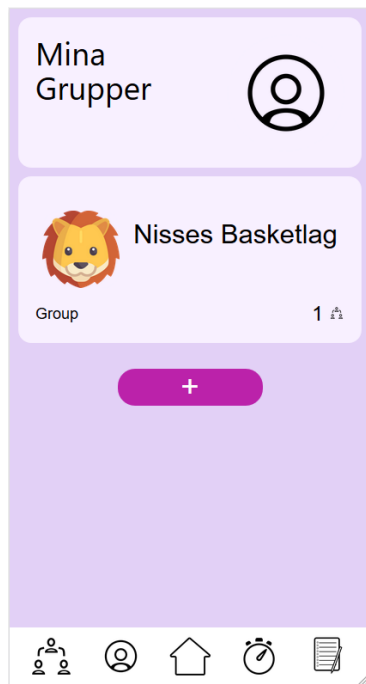
Figur 19: Detaljerad sida med info om en medlem i gruppen.



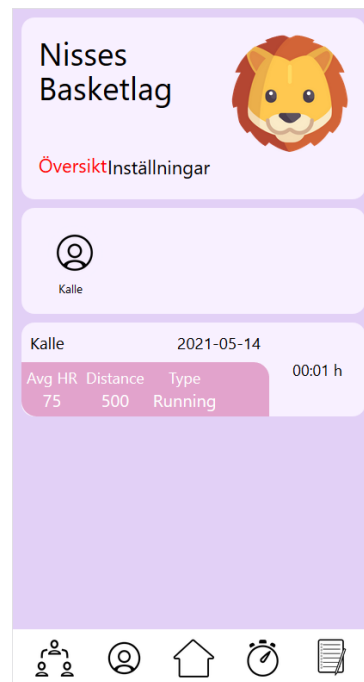
Sidan som visar grupper för att användare som inte är medlem i någon grupp.



Sidan för att skapa en ny grupp. Nås via pluss-knappen på bilden till vänster.



Sidan som visar grupper med gruppen skapad från föregående bild.



Detaljerad sida över gruppen från den nyskapade gruppen.



Profilsida där användaren kan ställa in mål.



Profilsidan där användaren kan ändra inställningar för sin profil.



The image shows a mobile application interface for a 'Profil' (Profile) settings page. At the top, there is a header with the title 'Profil' and two tabs: 'Mål' (Goals) and 'Inställningar' (Settings), with 'Inställningar' being the active tab. Below the header, there are several input fields for user information: 'Namn:' (Name) with the value 'frida', 'Användarnamn:' (Username) with 'friday', 'Mejl:' (Email) with 'saturday', 'Lösenord:' (Password) with 'sunday', 'Födelsedatum:' (Date of birth), 'Längd:' (Height), and 'Vikt:' (Weight). At the bottom of the form, there are two buttons: 'Importera data' (Import data) and 'Exportera data' (Export data). A bottom navigation bar contains five icons: a group of people, a person profile, a house, a clock, and a document.

Profil

Mål Inställningar

Namn:
frida

Användarnamn:
friday

Mejl:
saturday

Lösenord:
sunday

Födelsedatum:

Längd:

Vikt:

Importera data

Exportera data

Icons: Group of people, Profile, Home, Clock, Document

Profilsidan där användaren kan
ändra inställningar för sin profil.

B Blankett för GDPR

Samtycke till behandling av personuppgifter

Genom denna blankett efterfrågar vi ditt samtycke till behandling av personuppgifter. Nedan följer en beskrivning av vad ditt samtycke till behandling av personuppgifter innebär.

Personuppgiftsansvarig

Samtycket begärs in av Chalmers tekniska högskola, 556479-5598. Chalmers är personuppgiftsansvarig för de personuppgifter som behandlas inom ramen för samtycket.

Vilka personuppgifter kommer att behandlas?

Exempelvis namn, personnummer, foto, hälsouppgifter, maxpuls, etc, telefonnummer, adress.

Vad är syftet med behandlingen av personuppgifter?

Hur kommer personuppgifterna att behandlas?

Hur länge kommer personuppgifterna att behandlas och sparas?

Återkallande av samtycke

Du kan när som helst återkalla ditt samtycke till att vi får behandla dina personuppgifter genom att kontakta magnus.karlsteen@chalmers.se.

Genom undertecknande bekräftas att jag har tagit del av ovanstående information och är införstådd med hur mina personuppgifter kan komma att behandlas. Jag ger mitt samtycke.

Datum: _____

Underskrift: _____

Namnförtydligande: _____

C Enkäter för studien

Hur var din träning?

Namn:

Hur intensiv kände du att träningen var?

	0	1	2	3	4	5	6	7	8	9	10	
Vila	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Maximal

Typ av träning:

Hur länge varade träningspasset (i min):

Hur är din energinivå just nu?

1	2	3	4	5	6	7	8	9	10
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Träningsdag:

- ☐ Måndag
- ☐ Tisdag
- ☐ Onsdag
- ☐ torsdag
- ☐ Fredag
- ☐ Lördag
- ☐ Söndag

Namn:

☐ Ja☐ Nej

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

○ ○ ○ ○ ○ ○ ○ ○ ○ ○

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○

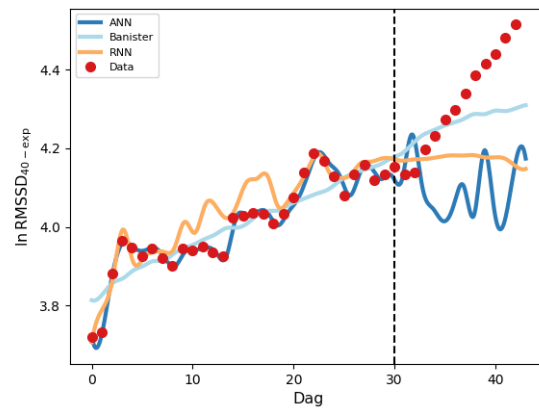
1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○

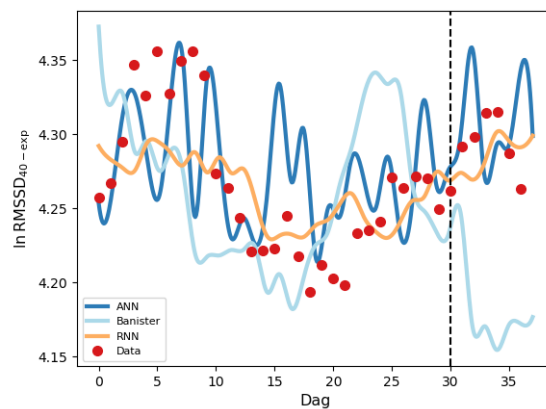
1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○

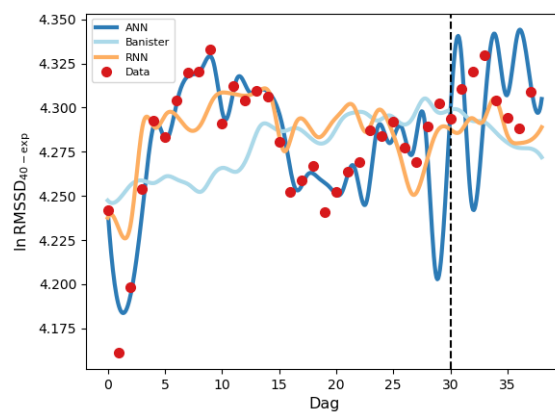
D Resultat



Figur 20: Figur över försöksperson #3 uppmätta $\ln \text{RMSSD}_{40} - \text{exp}$ plottad tillsammans alla tre modellerna Banister, ANN och RNN.



Figur 21: Figur över försöksperson #6 uppmätta $\ln \text{RMSSD}_{40} - \text{exp}$ plottad tillsammans alla tre modellerna Banister, ANN och RNN.



Figur 22: Figur över försöksperson #1 uppmätta $\ln \text{RMSSD}_{40} - \text{exp}$ plottad tillsammans alla tre modellerna Banister, ANN och RNN.

D.1 Härledning

Härledningen visas för fitness faktorn $g(t)$, men samma härledning gäller också för trötthetsfaktorn.

Fitness faktorn uttrycks som

$$g(t) = g(t-i)e^{-i/\tau_1} + w(t). \quad (22)$$

Sätt $i = 1$ och $w(t) = T$. Då gäller

$$\begin{aligned} g(t) &= g(t-1)e^{-1/\tau_1} + T \\ g(t) &= e^{-1/\tau_1} \left[g(t-2)e^{-1/\tau_1} + T \right] + T \dots \\ g(t) &= T \left[1 + e^{-1/\tau_1} + e^{-2/\tau_1} + \dots + e^{-(t-1)/\tau_1} \right] \\ &= T \sum_{i=1}^{t-1} e^{-(t-i)/\tau_1}. \end{aligned} \quad (23)$$

