UPPSALA
UNIVERSITET

UPTEC F 23055

Examensarbete 30 hp

September 2023

Club Head Tracking

Visualizing the Golf Swing with Machine Learning

Fredrik Herbai

Civilingenjörsprogrammet i Teknisk Fysik

UPPSALA
UNIVERSITET

Club Head Tracking
Visualizing the Golf Swing with Machine Learning

Fredrik Herbai

Abstract

During the broadcast of a golf tournament, a way to show the audience what a player's swing looks like would be to draw a trace following the movement of the club head. A computer vision model can be trained to identify the position of the club head in an image, but due to the high speed at which professional players swing their clubs coupled with the low frame rate of a typical broadcast camera, the club head is not discernible whatsoever in most frames. This means that the computer vision model is only able to deliver a few sparse detections of the club head.

This thesis project aims to develop a machine learning model that can predict the complete motion of the club head, in the form of a swing trace, based on the sparse club head detections.

Slow motion videos of golf swings are collected, and the club head's position is annotated manually in each frame. From these annotations, relevant data to describe the club head's motion, such as position and time parameters, is extracted and used to train the machine learning models. The dataset contains 256 annotated swings of professional and competent amateur golfers.

The two models that are implemented in this project are XGBoost and a feed forward neural network. The input given to the models only contains information in specific parts of the swing to mimic the pattern of the sparse detections.

Both models learned the underlying physics of the golf swing, and the quality of the predicted traces depends heavily on the amount of information provided in the input. In order to produce good predictions with only the amount of input information that can be expected from the computer vision model, a lot more training data is required. The traces predicted by the neural network are significantly smoother and thus look more realistic than the predictions made by the XGBoost model.

Teknisk-naturvetenskapliga fakulteten

Uppsala universitet, Utgivningsort Uppsala/Visby

Handledare: Alexandros Papachristou Ämnesgranskare: Ping Wu

Examinator: Tomas Nyberg

Populärvetenskaplig sammanfattning

Den som tittar på golf på TV får se de bästa spelarna i världen och hur deras golfsvingar ser ut. Trots att man får se en spelares sving kan det vara svårt att uppfatta hur klubbhuvudet egentligen har rört sig, delvis på grund av hur snabbt spelarna svingar. Ett sätt att göra det lättare för den som tittar att se hur klubbhuvudet har rört sig är att rita ut ett spår efter klubbhuvudet som visar var det har befunnit sig.

Man kan använda bildanalys för att lokalisera klubbhuvudet i en bild, men då måste klubbhuvudet vara synligt. Tyvärr gör kombinationen av spelarnas höga svinghastighet och kamerornas låga bildhastighet att klubbhuvudet inte går att urskilja i majoriteten av bilderna. Med andra ord kan man med hjälp av bildanalys bara lokalisera klubbhuvudet i några få bilder. Dessa lyckade detektioner sker i de delar av svingen då klubbhuvudet rör sig tillräckligt sakta.

Syftet med det här examensarbetet är att utveckla en maskininlärningsmodell som kan förutse klubbhuvudets kompletta rörelse utifrån några få kända positioner.

Slow motion-videor, där klubbhuvudet går att urskilja i nästan varje bild, används för att generera data. 256 slow motion-videor på golfsvingar samlas in och klubbhuvudets position annoteras manuellt i varje bild. Relevanta parametrar, som position och tid, extraheras ur annoteringarna och används för att träna maskininlärningsmodellerna. De två modellerna som används är XGBoost och ett neuralt nätverk. XGBoost är en ensemblemetod som kombinerar förutsägelser från flera beslutsträd och neurala nätverk efterliknar i viss grad strukturen i en människas hjärna. Informationen som ges till modellerna är strukturerad enligt klubbhuvudets synlighet i en vanlig video. Det vill säga information om klubbhuvudet ges i de långsamma delarna av svingen.

Modellerna har lärt sig den underliggande fysiken bakom golfsvingen och kan, givet tillräcklig mycket information om de långsamma delarna av svingen, leverera goda förutsägelser. Hur korrekta förutsägelserna är alltså starkt beroende av mängden information modellerna får om svingen. För att ge tillfredställande förutsägelser givet antalet förväntade klubbhuvudsdetektioner behöver modellerna mer träningsdata. Generellt är det neurala nätverkets förutsagda svingar slätare och därmed mer realistiska än de förutsagda av XGBoost.

Acknowledgements

I would like to express my gratitude toward all those who helped me during the course of this thesis. Without them this work would not have been possible.

First of all, I want to thank my supervisor Alexandros for offering his guidance and support throughout this project. He assisted me with everything from the data collection process to offering key insights in machine learning.

In addition, I want to thank all those who allowed me to record their golf swings, thus providing invaluable data for the machine learning models.

Finally, I want to thank my subject reader Ping, who guided me through the process of conducting the project and helped me a great deal with writing this report.

Contents

1 Introduction

1.1 Background

During the broadcast of a golf tournament, the broadcaster may want to show the audience how the club moves throughout a player's swing. This could be achieved by tracking the club head's movement and producing a visual trace of the path that it has traveled. An example of what swing trace could look like is show in Figure 1.1. A golf swing can be divided into three main parts; *the backswing*, where the club is taken back, *the downswing*, where the club returns to strike the ball, and finally *the follow through*, which constitutes the rest of the swing after impact with the ball. The red curve shows the trace of the backswing and the blue curve shows the trace of the downswing. The player has just started the follow through.

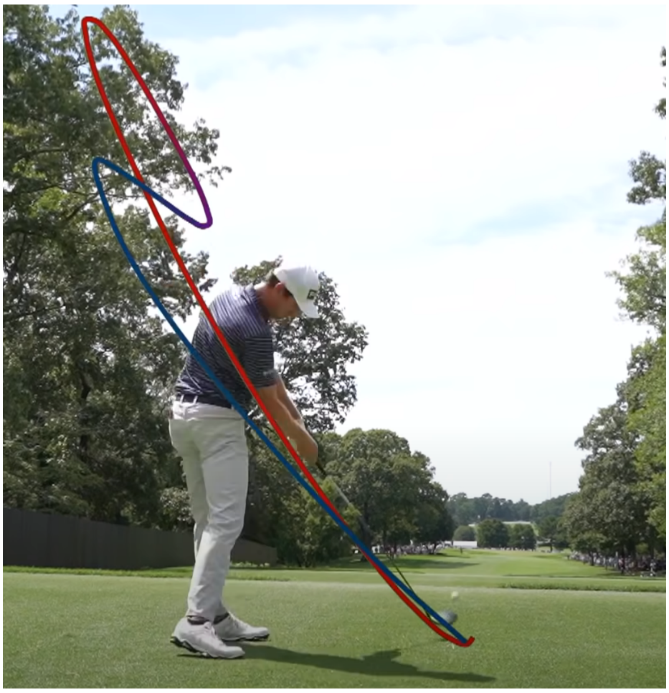


Figure 1.1: Example of a swing trace [1]

However, tracking the movement of the club head is not trivial. Professional golfers swing their clubs at incredibly high speeds, with some reaching club head speeds of over 200 km/h with their drivers. Meanwhile, the typical broadcast camera used at a golf tournament has a frame rate of 30 fps (frames per second) or 60 fps interlaced. The high club head speed coupled with the low frame rate broadcast cameras means that the club head travels a long way in between frames. As a result the club head is not visible in most frames. When it is visible, the resulting image will likely depict a blurry instance of the club head. A couple of frames from a video recorded at 30 fps are presented in Figure 1.2 to illustrate the problem of locating the club head in the downswing. Note that the club head is visible in a) which shows the top of the swing, i.e. the end of the backswing. As soon as the downswing is initiated the club head accelerates rapidly and is not visible again until the middle of the follow through seen in i). Upon reaching the end of the follow through the club head has slowed down enough to be visible in m), n) and o), as well as all the subsequent frames following o) which are not included in this example.

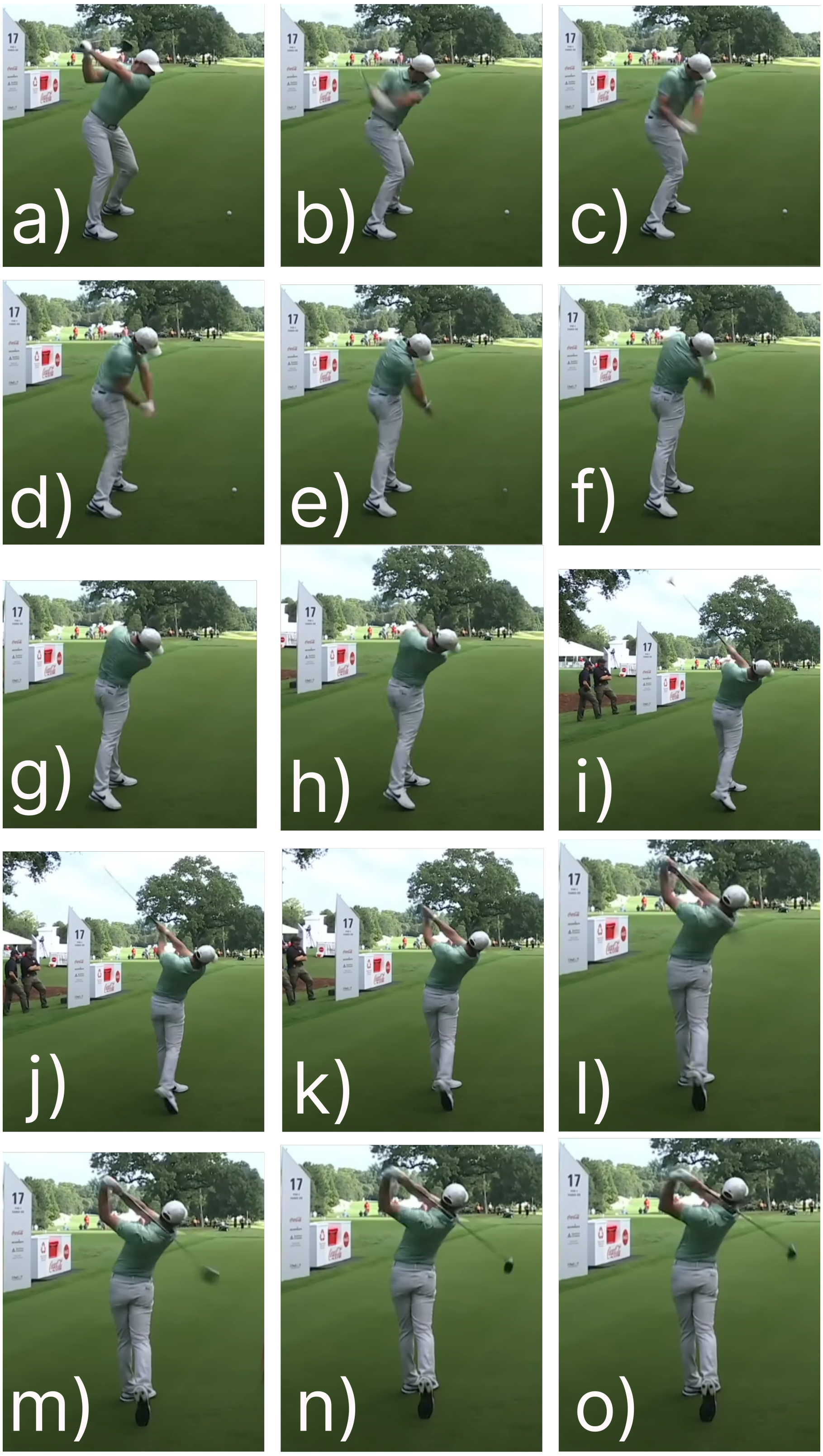


Figure 1.2: Club head visibility in the frames of a video recorded at 30 fps. The club head is visible in subfigures a), i), m), n) and o) [2]

Fortunately, the club head is not always traveling at 200 km/h. It speeds up and slows down throughout the swing. In the slow-moving parts of the swing, i.e. the beginning of the backswing, the top of the swing, and the end of the follow through, the broadcast camera is able to capture some clear images of the club head. Still, a computer vision model able to detect the head of a golf club in an image would only be able to deliver very few detections. It would be able to do this for only a couple of frames in the slow-moving parts of the swing and the vast majority of all the information about the club head's position and movement would be lost.

1.2 Purpose and Goals

The goal of this project is to build and train a machine learning model to learn the underlying patterns of the physics of the golf swing so that it can take these sparse detections as input and predict the complete motion of the club head, represented as a trace of its path.

1.3 Tasks and Scope

In order to achieve the goals, this master thesis project is conducted particularly with the following tasks:

1. study the kinematics of the golf swing;
2. study and investigate existing machine learning techniques and models, and determine which ones are best suited to the project;
3. collect slow motion video data of golf swings, in which the position of the club head is annotated in each frame. Relevant data to describe the club head's motion will be extracted from the annotations and it will subsequently be used by machine learning models to learn the patterns in the data;
4. implement the machine learning models and evaluate their performance.

1.4 Outline

The remainder of this thesis is structured in the following way. Section 2 presents some of the kinematics in golf and provides a motivation for visualising the golf swing as a trace. Section 2 also describes the theory behind the machine learning models used in this thesis. In section 3 the process of collecting and annotating data is explained. It also shows how the data is pre-processed and formatted, as well as how the machine learning models are implemented. The results are presented and discussed in section 4. Finally, the conclusions of the thesis are presented in section 5 along with some suggested improvements to the models.

2 Theory

2.1 Kinematics of the Golf Swing

The purpose of this section is to explain and motivate, especially to non-golfers, why the club head's motion throughout the golf swing is of interest.

2.1.1 Golf Clubs

There are five different categories of golf clubs: *woods*, *irons*, *wedges*, *hybrids*, and *putters*. Due to the differences in construction, a player's swing will be slightly different depending on what category of club they are using. In this thesis the focus will be on woods and irons since they are the two most distinct categories (except putters).

Woods get their name from the material that was originally used to construct them. Nowadays woods are typically made of titanium and carbon. The features that characterize woods are the large and hollow club heads as well as the long graphite shafts. The increased shaft length and the light weight allow players to swing the club faster. Woods are designed to help players launch the ball higher with more ball speed and backspin and thus they occupy the lowest lofted part of the bag. Examples are drivers and fairway woods. As the name implies fairway woods can used both when the ball lies on the ground and when it is teed up on a short tee peg. In general fairway woods should be hit with a slightly negative angle of attack. Drivers on the other hand are specifically designed to be used when the ball is teed up on a tall tee peg allowing the player to hit the ball with a positive attack angle. The positive/slightly negative angle of attack as well the club head travelling further in the backswing due to the longer shaft are the main visual differences in the golf swing when using a wood.

Irons on the other hand have small compact club heads which are usually made of solid forged steel making them much heavier than woods. They come in numbered sets that occupy the middle part of the bag. The higher that number is, the more loft is on the club and the shorter the shaft is. Since players typically launch the ball high enough using these medium to high lofts, the design can instead emphasize control. Irons should always be used to hit the ball from the ground or a from a very short tee. Thus, the angle of attack should always be negative. In fact, the shorter the iron is, the more negative the angle of attack should be. The main visual difference in the golf swing when using an iron is that the club head does not travel as far in the backswing and that the atack angle is more negative.

Wedges are effectively shorter, higher lofted, and more specialized irons. Hybrids are a mix between irons and woods. And finally, putters are used to roll the ball on the green.

2.1.2 Swing Kinematics

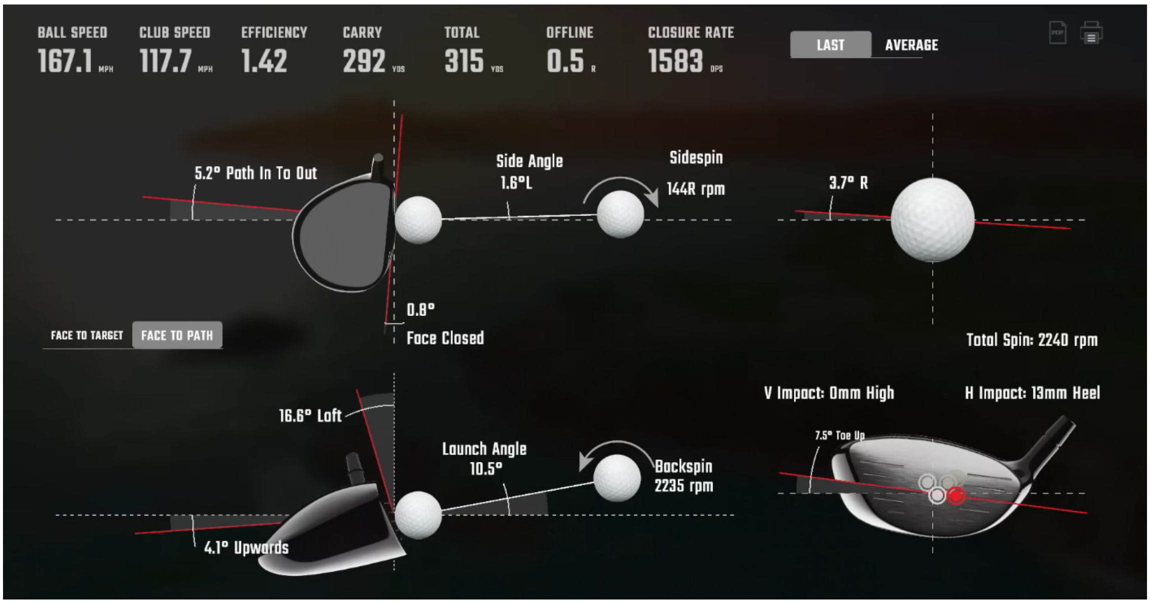
The way a player swings the club affects how the club head is delivered to the ball, which in turn is what determines the trajectory of the golf ball. The delivery at impact can be described by the following parameters:

- *club head speed* - how fast the club head is travelling;
- *path* - how much the club head is travelling left or right;
- *angle of attack* - how much the club head is travelling up or down;
- *dynamic loft* - how much the clubface is pointing upward;
- *face angle* - how much the clubface is pointing right or left;
- *strike location* - the location on the clubface which makes contact with the ball.

The resulting launch conditions of the golf ball can be described by the following parameters:

- *ball speed* - how fast the ball is travelling;
- *spin rate* - consists of both *backspin* and *sidespin*;
- *start direction* - how much the ball starts to the left or right of the target;
- *launch angle* - how high the ball launches.

A visualisation of the delivery and launch parameters when using driver is presented in Figure 2.1.



BALL SPEED	CLUB SPEED	EFFICIENCY	CARRY	TOTAL	OFFLINE	CLOSURE RATE
167.1 MPH	117.7 MPH	1.42	292 YDS	315 YDS	0.5 R	1583 DPS

Figure 2.1: Example of a driver delivery and the resulting launch conditions [3]

What is considered optimal in terms of launch conditions varies significantly depending on which club is being used (the difference is mainly in ball speed, backspin, and launch angle). For simplicity we will therefore consider the longest club in the bag, the driver. Due to the low loft, inefficiencies in delivery will be penalized more severely with the driver than with other clubs. That makes it a good candidate for this example. The goal with the driver could be summarized as "hitting the ball as far as possible while maintaining control of direction". In order to produce launch conditions that achieve this goal, certain things are required in the delivery. Namely, striking the ball near the middle of the clubface and aligning the face angle with the club path allows for an efficient transfer of kinetic energy from the club head to the ball, producing more ball speed and less sidespin. If the face angle deviates significantly from the path, a large portion of the kinetic energy of the club head will be transferred to the ball as sidespin instead of ball speed, causing excess curvature and reducing distance. It is also important to have a reasonable relationship between delivered loft and angle of attack in order to produce appropriate launch angle and backspin. Backspin generates a lift force proportional to the rate of backspin and the speed of the ball relative to the air. Too much backspin will generate too much lift, causing the ball to rise excessively. Too little backspin will not produce enough lift and the ball will consequently fall out of the air. Both scenarios reduce driving distance.

This brings us to what is probably the most common swing fault among inexperienced and unskilled golfers, known as the *over the top move*. Figure 2.2 shows an illustration of what the *over the top move* can look like. It involves the golfer pushing the club out in front of them in the downswing, causing them to come steeply into the ball. This produces an excessively leftward path along with a downward angle of attack and is usually accompanied with a face angle pointing way to the right of the path as well as an increase in dynamic loft. The face angle pointing way right of the leftward path produces lower than ideal ball speed and too much sidespin. The downward angle of attack combined with the high dynamic loft produces excessive backspin. The result is a weak ball-flight that curves off to the right; the antithesis of the previously defined goal of driving.

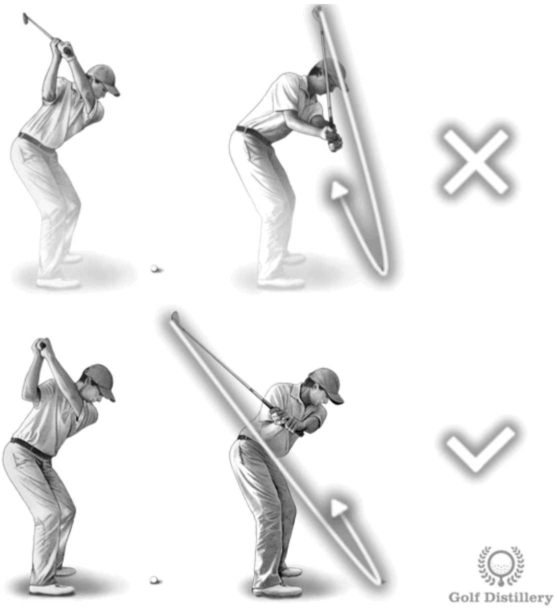


Figure 2.2: Illustration comparing the *over the top move* (top) with a more neutral downswing (bottom) [4]

Utilizing the proper kinematic sequence is required to produce a sound and efficient golf swing [5]. This refers to the way energy is transferred from one body segments to the next.

In the downswing, the pelvis is the first segment to reach its peak rotational speed. It then decelerates as energy is transferred to the thorax which reaches an even higher peak rotational speed [6]. The thorax decelerates as it transfers its energy into the arms which again reach an even higher peak rotational speed. Finally, the arms decelerate as their energy is transferred into the club. In linear terms, this final step means that the hands are slowing down in order for the club head to reach its maximal speed as it makes contact with the ball. The order and timing of when each segment reaches its peak rotational speed are presented in Figure 2.3.

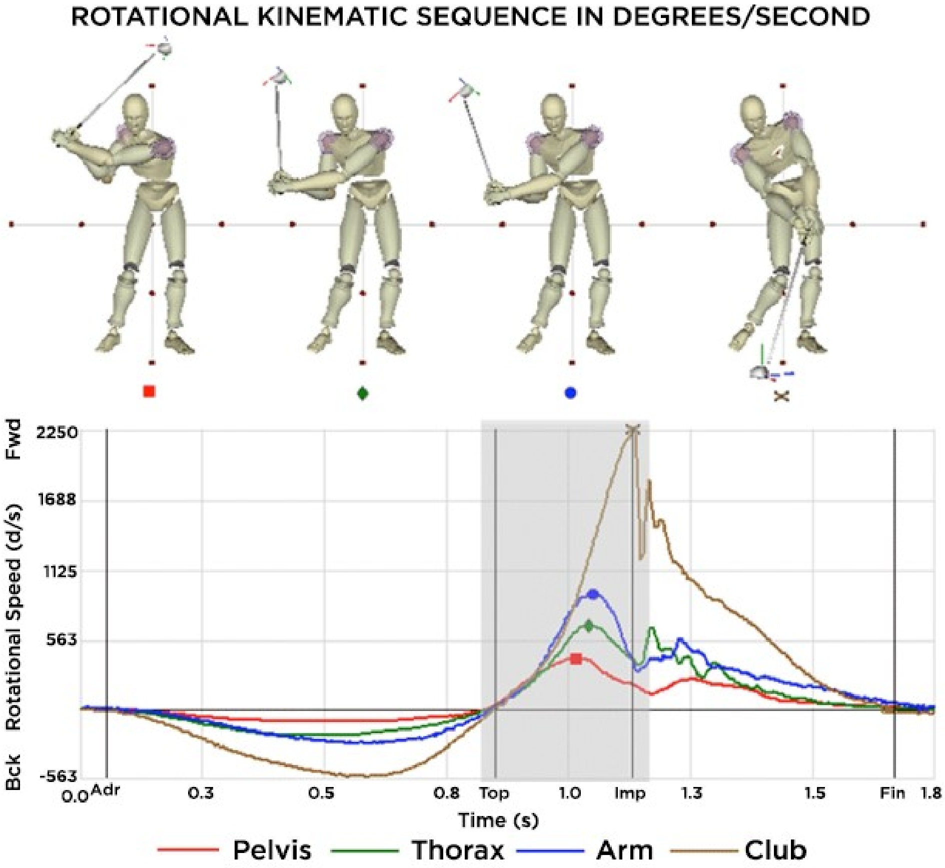


Figure 2.3: The rotational kinematic sequence of a world class golfer showing the positions and timings of the peak rotational speed for each segment [7]

Each segment builds upon the previous one, increasing the speed up the chain. Even though professional players' swings can vary significantly in appearance, all great ball strikers have the same sequence of energy generation in their golf swings [5, 6].

2.2 Machine Learning

Machine learning (ML) is a branch of *artificial intelligence* (AI) where machines learn and identify patterns in data. It can be split into four categories; *supervised learning*, *unsupervised learning*, *semi-supervised learning*, and *reinforcement learning*. This thesis will focus on supervised learning, which involves a model learning the relationship between input and output through labelled training data. Supervised learning can itself be split into two sub-categories: *classification* - used to predict discrete values, and *regression* - used to predict continuous values.

Two different machine learning models are deployed in this thesis, XGBoost and neural network (NN). XGBoost has peformed well in many machine learning competitions [8] and is considered a state of the art model due to its speed, efficiency, and prediction accuracy. NNs can be described as the go to models for learning complex patterns in data, but a NN typically requires a lot of training data to do this successfully. If the NN has more than two layers, the structure becomes complex enough to achieve *deep learning*.

2.2.1 XGBoost

XGBoost (extreme gradient boosting) is an ensemble method based on gradient boosting decision trees. Ensemble methods combine the predictions from multiple weak models to create a strong and more precise model. For XGBoost, the weak learners are classification and regression trees (CART).

A CART is a tree-like structure consisting of decision nodes and leaf nodes. An example of such a structure is shown in Figure 2.4. At the decision nodes, the data is recursively split into two subsets (branches) based on its features until a stopping criterion is met. A stopping criteria could be that a minimum number of datapoints remain in a node. Nodes where the recursion stops are called leaves. The split conditions are selected to maximize the information gain of the current split. Therefore the splits are referred to as greedy, since they do not take future splits into account. Deep CARTs, which have a large amount of sequential splits, are prone to overfitting the training data, while shallow CARTs, which have a small amount of sequential splits, often fail to learn important patterns.

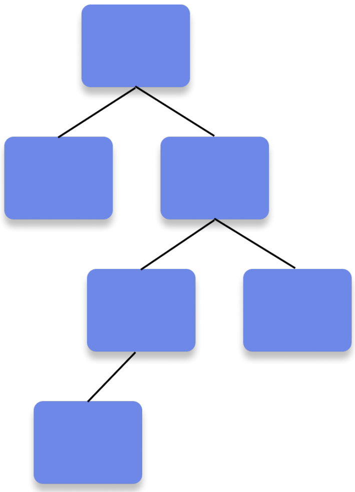


Figure 2.4: Example of a CART structure

Gradient boosted trees is an ensemble model where trees are built sequentially, one at a time. Each new tree is constructed with the previous prediction error in mind so that the trees complement each other. The final prediction takes into account the predictions from all trees [9]. Such a collective prediction can be described mathematically as

$$\hat{y}_i = \sum_{k=1}^{K} f_k(\mathbf{x}_i), \quad f_k \in \mathcal{F}, \tag{2.1}$$

where $\hat{y}_i$ is the collective prediction, K is the number of trees, f_k is a function representing an individual tree, x_i is an input datapoint, and $\mathcal{F}$ is the function space containing all possible CARTs. Each leaf node of a CART has an associated weight, w_i, which will be provided below. The dataset is distributed according to the decision rules of the tree structure and the tree's prediction is calculated by adding up its weights [10].

The model's objective function to be optimized is defined as

$$\mathcal{L} = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(f_k). \tag{2.2}$$

The first term is the loss function which measures the difference between the ground truth y_i and the model's prediction $\hat{y}_i$. The second term is the regularization term where $\Omega(f)$ is the complexity of the tree f, which is defined as

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda ||w||^2. \tag{2.3}$$

The regularization term helps to reduce over-fitting by penalising tree complexity [10].

The objective function cannot be optimized using traditional optimisation methods since it includes functions as parameters. It must instead be trained in an additive manner [10]. That which has already been learned remains and a new tree is added to the model in each step. Thus, the prediction at step t can be written as

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(\mathbf{x}_i) = \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i). \tag{2.4}$$

The f_t which is added to the model is selected greedily such that the objective function,

$$\mathcal{L}^{(t)} = \sum_i^n l(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)) + \Omega(f_t), \tag{2.5}$$

is minimized at the current step [9]. Applying a second order Taylor expansion yields an approximate objective function

$$\mathcal{L}^{(t)} \simeq \sum_{i=1}^{n}[l(y_i, \hat{y}^{(t-1)}) + g_i f_t(\mathbf{x}_i) + \frac{1}{2}h_i f_t^2(\mathbf{x}_i)] + \Omega(f_t), \tag{2.6}$$

where $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ and $h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ are first and second order gradient statistics on the loss function [10]. By removing the constant terms, a simplified objective function at step t can be obtained,

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^{n}[g_i f_t(\mathbf{x}_i) + \frac{1}{2}h_i f_t^2(\mathbf{x}_i)] + \Omega(f_t). \tag{2.7}$$

By expanding Ω, (2.7) can be written as

$$\begin{aligned} \tilde{\mathcal{L}}^{(t)} &= \sum_{i=1}^{n}[g_i f_t(\mathbf{x}_i) + \frac{1}{2}h_i f_t^2(\mathbf{x}_i)] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \\ &= \sum_{j=1}^{T}[(\sum_{i\in I_j} g_i)w_j + \frac{1}{2}(\sum_{i\in I_j} h_i + \lambda)w_j^2] + \gamma T, \end{aligned} \tag{2.8}$$

where I_j is the set of indices to the data points assigned to leaf j. The indexation is updated in the second row of (2.8) because all the data points on the same leaf get the same score [9]. The optimal weight w_j^* of leaf j in a fixed tree structure can be computed by

$$w_j^* = -\frac{\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h_i + \lambda}, \tag{2.9}$$

and the corresponding optimal value of the objective function for the fixed tree structure can be calculated by

$$\tilde{\mathcal{L}}^{(t)} = -\frac{1}{2}\sum_{j=1}^{T} \frac{(\sum_{i\in I_j} g_i)^2}{\sum_{i\in I_j} h_i + \lambda} + \gamma T. \tag{2.10}$$

This value can be used to measure the quality of a tree structure in a similar manner to an impurity score. Since it is almost always infeasible to go through all of the possible tree structures, a greedy algorithm is used instead [10]. It starts from a single leaf and iteratively adds branches. The loss reduction after a split is given by

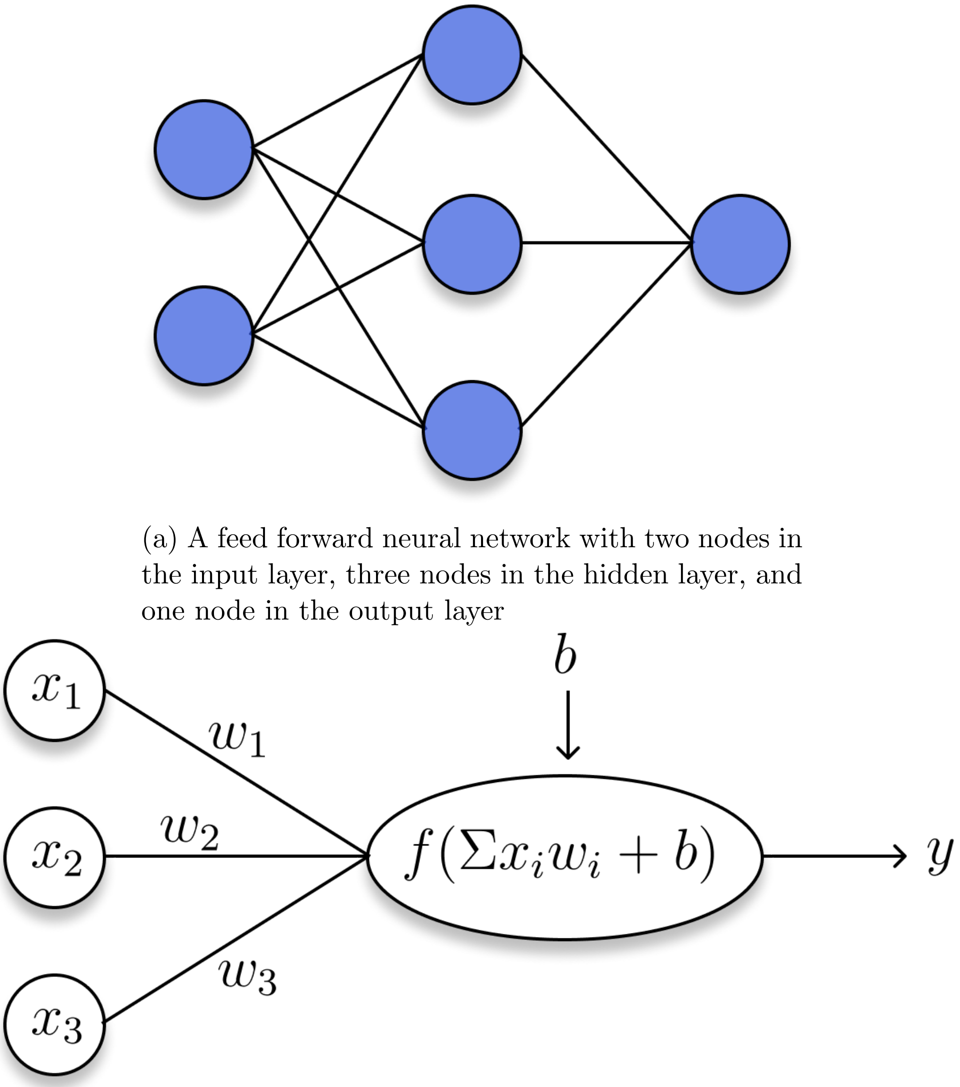
$$\mathcal{L}_{\text{split}} = \frac{1}{2}\left[\frac{(\sum_{i\in I_L} g_i)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{(\sum_{i\in I_R} g_i)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{(\sum_{i\in I_j} g_i)^2}{\sum_{i\in I_j} h_i + \lambda}\right] - \gamma, \tag{2.11}$$

where I_L and I_R are the sets of indices of the left and right nodes after the split. The terms in (2.11) can be understood as the score of the two new leaves, the score on the original leaf, and the regularization of the new leaf [9].

XGBoost uses this exact principle. It is an implementation of the gradient boosted trees algorithm pushed to the extreme in terms of system optimisation and computation limits.

2.2.2 Neural Networks

Neural networks (NN) loosely mimic the human brain. They consist of a series of interconnected layers of nodes. The nodes fire, like neurons in a brain, depending what input they receive [11]. This is called an activation and is represented by a numerical value. In the most basic version, known as a *feed forward neural network*, the nodes in a given layer are connected to all the nodes in the subsequent layer. It is called Feed Forward because all information travels in one direction; from the input layer to the output layer. The layers in between the input and output are called hidden layers. Figure 2.5 shows what a small NN can look like.

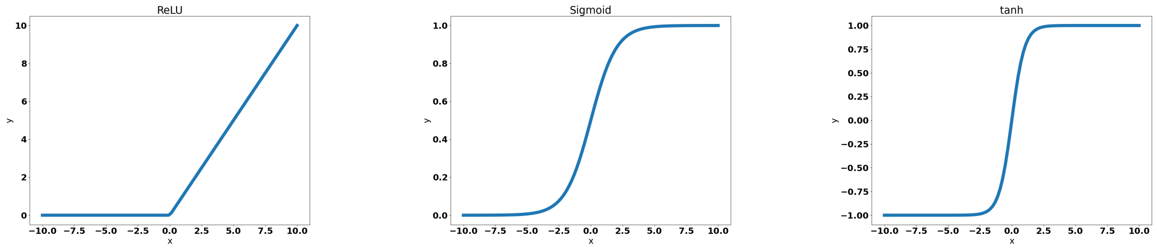


(a) A feed forward neural network with two nodes in the input layer, three nodes in the hidden layer, and one node in the output layer

(b) How the output of a node is determined

Figure 2.5: Example structure of a small NN and a visualization of how the output of each node is determined

Each connection has an associated weight w and each node has an associated bias b. The activation of a node is determined as follows: The node receives the activation values from the previous layer as input. Each activation value is individually multiplied with the weight of its connection. The products are all added together along with the node's bias. The sum is finally passed through an activation function which ultimately decides whether or not, and how much, the node will activate [12]. Some commonly used activation functions include the Rectified Linear Unit function (ReLU), the Sigmoid function, and the Hyperbolic Tangent function (tanh). These functions are depicted in Figure 2.6.



(a) ReLU

(b) Sigmoid

(c) tanh

Figure 2.6: Graphs of three common activation functions

The output y of a node is thus given by

$$y = f\left(b + \sum_{i=1}^{m} x_i w_i\right), \tag{2.12}$$

where f is an activation function, and m are the number of inputs x from the previous layer.

These operations are performed for all nodes in the network until the information reaches the output layer. There, the output can be transformed to the desired format. For example, classification or regression.

The values of the network's parameters, the weights and biases, are adjusted when training the network. Deciding how the parameters should be adjusted is done through a process know as *backpropagation* [12]. When the NN receives an input from the training data, its resulting output is compared to the ground truth. The error is evaluated through a *loss function*, for example the mean squared error (MSE) or the mean absolute error (MAE). The backpropagation algorithm calculates the negative gradient of the loss function with respect to the weights and biases, i.e. the direction in which to adjust the parameters to achieve the greatest reduction in the loss function. This is done by applying the chain rule recursively to each layer in the backward direction, from the output layer to the input layer. An optimisation algorithm then uses the gradient of the loss function to adjust the parameters accordingly. The backpropagation determines in what direction to make an adjustment, and the optimisation algorithm determines how large that adjustment should be. Examples of common optimisation algorithms are Stochastic gradient descent (SGD) and Adam.

3 Methods and Implementation

3.1 Hardware and Setup

3.1.1 Smartphones

In person videos are recorded using the slow motion feature on mobile phone devices. Two phones are used.

Table 3.1: Specifications of mobile phone devices used to record slow motion videos.

Mobile phone device	resolution	frame rate
Samsung Galaxy S8	720p	240 fps
Samsung Galaxy S22	1080p	240 fps

3.1.2 Setup

When recording a player's swing, it is imperative that the camera is positioned at an adequate distance from the player to ensure that the club head does not leave the frame at any point during the swing. If it does, the video can not be annotated correctly. To achieve this the camera is positioned approximately 5-6 m behind the golfer. At that distance the camera is moved between recordings at about 0.5 m increments in a horizontal span of roughly 4 m. Meaning that multiple videos of the same player are recorded from various angles.

3.2 Data Collection

Slow motion videos of golf swings captured with a high frame rate camera are used to generate training data. In these videos the club head is discernible in almost every frame. Data is generated by annotating the club head position in each frame of a video. x- and y-positions for different points in time are extracted from the annotated pixel and the time parameter is derived from the frame number. The data for one swing therefore consists of a time series of x and y coordinates.

The videos used for generating the dataset include both professional and amateur golfers. Some of the videos are collected from the internet and some are recorded in-house.

The camera perspective of interest in this project is known as *down the line*. An example of this perspective is shown in Figure 3.1. The dataset contains down the line videos of a wide variety of players performing full golf swings using both irons and woods. A slight change in camera position produces a different looking trace for the same player's swing. Therefore, multiple videos of the same player can be included in the dataset without increasing the risk of overfitting. This fact is utilized to the fullest when recording videos, by capturing the same player's swing from multiple camera positions (while always keeping the down the line perspective). For simplicity, only right handed golfers are included in the dataset. A model trained only on right handed golfers can still be used to make predictions on the swing of a left handed golfer by for example, horizontally mirroring the video.

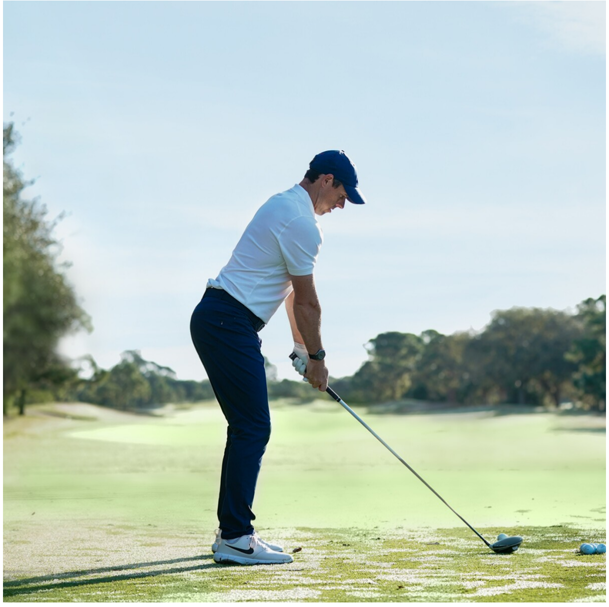


Figure 3.1: Down the line perspective of a golfer addressing the ball [13]

In total, the dataset includes 256 swings divided into 80% training, 10% validation, and 10% test data. 168 of the swings are captured in-house and 88 of them are collected from the internet.

For each golfer, this procedure is performed once with a driver or three wood, and once with a mid-iron.

For each player that participated in the in-house recordings, a series of swings recorded from the varying camera positions described in 3.1.2 was collected once with a driver and once with a mid-iron.

The simulator studio where most of the videos are recorded is quite dark and the woods used in the recordings are matte black. Bright yellow tape is therefore added to the club head of the woods in order to improve visibility in the recordings. This is not necessary with the irons since they already have a bright and shiny surface.

3.3 Data Pre-Processing and Format

The parameters that describe the club head's position in space and time are normalized and rescaled so that all swings exist in a standard domain with side length of 100. The initial position of the club head (placed just behind the ball at address) constitutes the origin of both the x- and y-coordinates as well as the time parameter. The highest position that the club head reaches along the y-axis is defined as 100 while the leftmost position along the x-axis is defined as -100. The time parameter is confined to the interval (0,100). The initial time is when the club moves away from the ball in the beginning of the backswing, and the final time is when it has moved to a halt in the end of the follow through.

For each swing, a regression model is trained on the time series and is used to generate x and y coordinates for 200 equidistant points in time. During training, these 200 points serve as the ground truth while the input to the model is a small selection of those same points. In other words, the model receives an incomplete time series as input and predicts what the complete time series looks like.

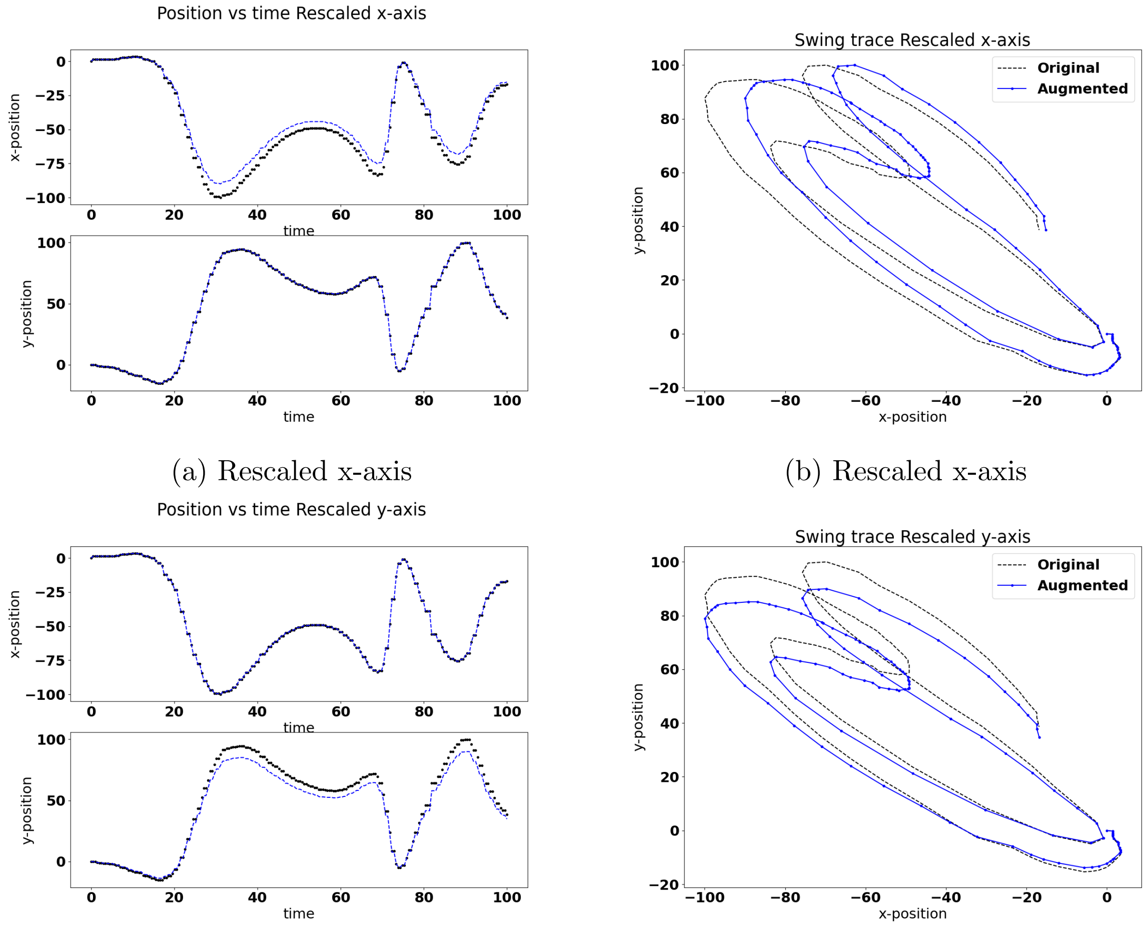
The input data is made to simulate detections delivered by the computer vision model. Steps taken to achieve this include, only including club head detections from the slow-moving parts of the swing, as well as varying the number of club head detections for different swings. Having variable length input to the models is handled by padding the input vectors with zeros.

3.4 Data Augmentation

Data augmentation is utilized in order to artificially increase the size of the dataset. Augmentation is only applied to the training data. Not to the validation or test data. Three augmentations are tested. Namely, Rescaling the axes, Contraction/Expansion of the trace, and slightly Perturbing the points randomly. The following figures compare what a trace looks like before and after different augmentations have been applied.

3.4.1 Rescaling

All the values of one of the coordinate parameters are slightly increased or decreased, creating a different looking trace. Figure 3.2 demonstrates what this augmentation can look like.



(a) Rescaled x-axis

(b) Rescaled x-axis

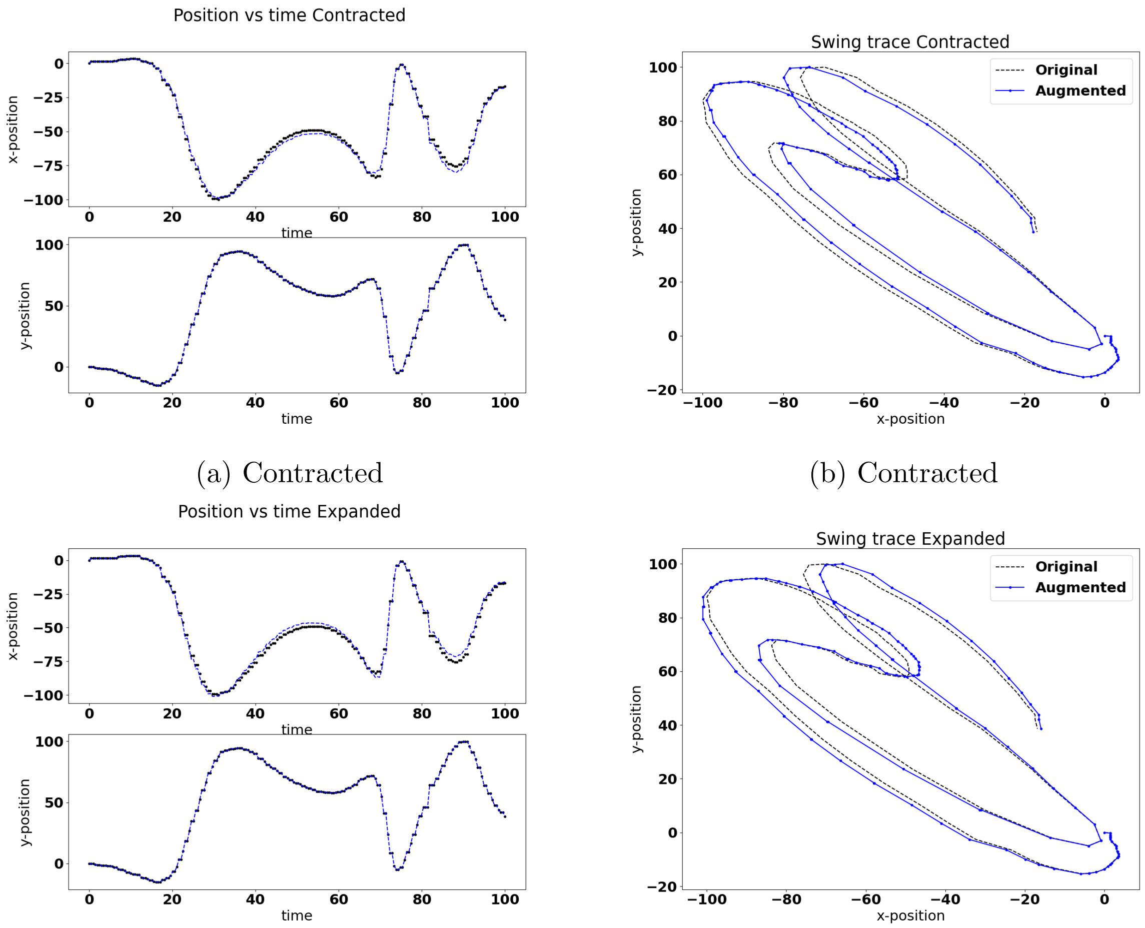
(c) Rescaled y-axis

(d) Rescaled y-axis

Figure 3.2: Comparison between a trace augmented through rescaling and the original trace

3.4.2 Contract and Expand

The trace is either contracted or expanded, as shown in Figure 3.3, in order to simulate a horizontal shift in the cameras position. The boundary between contraction and expansion is selected naively by considering the most common points in time where the club head transitions from the near side to the far side of the player and vice versa. Points are affected less and less by the contraction/expansion the closer to the transition region they are.



(a) Contracted

(b) Contracted

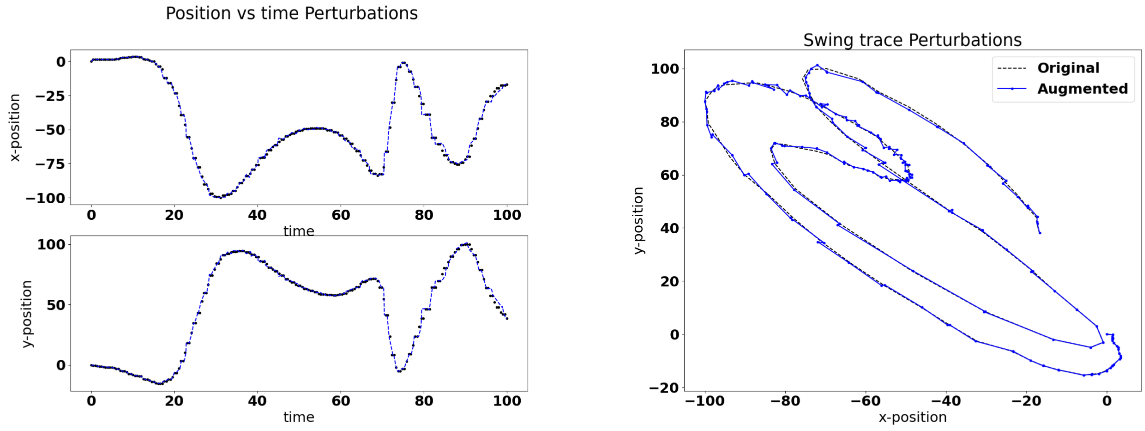
(c) Expanded

(d) Expanded

Figure 3.3: Comparison between a trace augmented through Contraction/Expansion and the original trace

3.4.3 Random Perturbations

Small random perturbations are applied individually to all values of the x, y, and time parameters. An example of the resulting trace is presented in Figure 3.4



(a) Perturbed

(b) Perturbed

Figure 3.4: Comparison between a trace augmented through Random perturbations and the original trace

3.5 Machine Learning Models

A model's parameters are the variables that change during the training process. Finding the best parameter values actually is the learning. A model also has *hyperparameters*, which are not affected by the training of the model. On the contrary, they are used to control the training. Examples of hyperparameters are *learning rate* or the number of hidden layers in a NN. Since the they impact training, it is important to try and optimize the hyperparameters so that the model achieves good performance. The process of finding the optimal hyperparameters to use is called *hyperparameter tuning*.

Both models use MSE as cost function. The formula of which is given by

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2. \tag{3.1}$$

Where y_i is the prediction and $\hat{y}_i$ is the ground truth of datapoint i and n is the number of datapoints.

The implementation details of XGBoost and the NN are presented below.

3.5.1 XGBoost

An XGBoost regression model is implemented in python using the *xgboost* library [14]. The model is tuned using *HyperOpt* [15] and the resulting hyperparameters are listed in Table 3.2.

Table 3.2: Tuned hyperparameters for the XGBoost regressor

alpha	1.1984
colsample_bytree	0.8022
gamma	72.365
lambda	0.2660
learning_rate	0.0586
max_depth	63.953
min_child_weight	9.6802
subsample	0.7848

3.5.2 Neural Network

A feed forward neural network is implemented in python using pytorch. The hyperparameters of the NN are tuned manually and the resulting specifications are listed in Table 3.3. Due to the limited training data, mainly shallow network structures with linear layers are considered. In this case shallow means networks with one, two, or three hidden layers. As for the hidden layer size, values from the 2^n series between 8 and 256 are considered. ReLu, tanh, and Sigmoid are tested as activation functions and Adam and SGD are tested as optimisation algorithms. When evaluating performance in tuning process, the MSE is of primary interest, but the visual appearance of the traces is considered as well.

Table 3.3: Implementation specifications of NN

Batch size	32
Number of hidden layers	2
Size of hidden layers	128
Activation function	ReLU
Optimisation Algorithm	Adam
Adam learning rate	10^{-3}
Adam weight decay	10^{-3}

Early stopping is utilized as a form of regularisation, cancelling the training process if the validation loss has not improved in the last 200 epochs. At which point the model reverts to the parameters that yield the best performance on the validation data.

4 Results and Discussions

4.1 Number of club head Observations

The models' predictive capabilities are tested for a varying number of club head observations, which will be called *sample size* for short. The average MSE for the different sample sizes is presented in a table. In addition, graphs are presented for each sample size comparing a prediction with the ground truth. Two different graphs are used for such a comparison. One showing the x- and y-coordinates against time and the other showing the resulting trace of the club head. For each sample size, a selection of predictions is presented, exemplifying what a good, an average, and a poor prediction can look like for that sample size. These selections are made trying to represent the models' behaviour as accurately as possible. In the graphs, the black data points are the ground truth, the blue are the predictions, and the red indicate the regions from which the model has received club head observations as input data. Additional average predictions for both models are presented in appendix A.

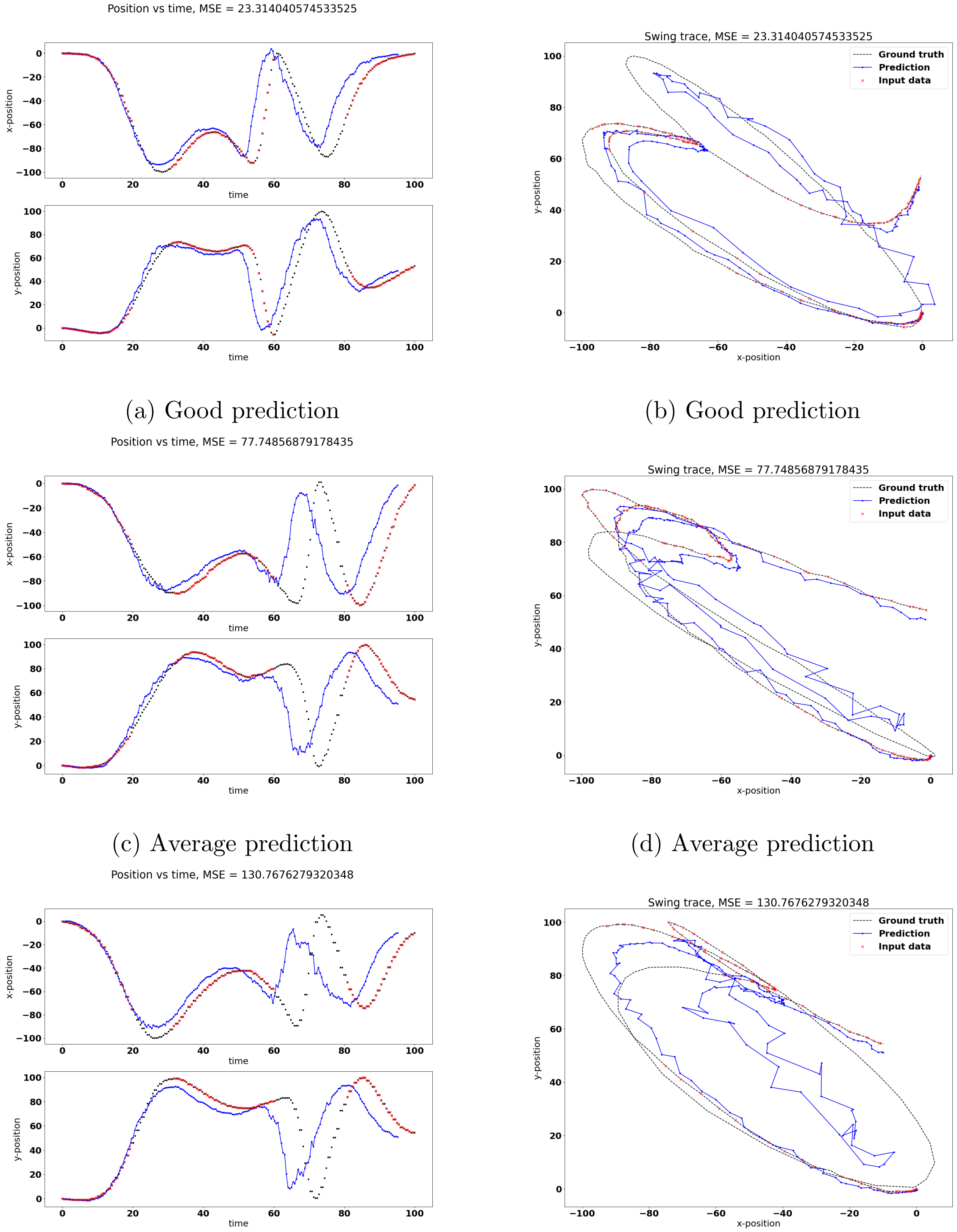
The Average MSE of XGboost's predictions on the testdata for different sample sizes is presented in Table 4.1. It is evident that the MSE increases as the model receives less information through the input. It also appears that the reduction in prediction accuracy is more drastic for lower sample sizes since the largest reduction, both in terms of values and proportion, takes place when going from 75 to 50 data points.

Table 4.1: Average MSE of testdata predictions by XGBoost for different sample sizes

Sample size	MSE
125	62.70
100	82.79
75	88.45
50	124.66

Figure 4.1, Figure 4.2, Figure 4.3, and Figure 4.4 present graphs of predictions made by XGBoost for different sample sizes. From the graphs on the left hand side, which show the x- and y-coordinates of the club head against time, it is clear that XGBoost has at least to some extent learnt the underlying pattern of the club head's movement in the golf swing. Visually there is a descent match between the prediction and the ground truth, although it deteriorates somewhat for the smallest sample sizes.

However, when examining the graphs on the right hand side, which show what the predicted traces look like, it is easy to see how noisy the predictions are. Quite a few of them do not even resemble club head movements at all because of how squiggly the lines are.



Position vs time, MSE = 23.314040574533525

Swing trace, MSE = 23.314040574533525

Position vs time, MSE = 77.74856879178435

Swing trace, MSE = 77.74856879178435

Position vs time, MSE = 130.7676279320348

Swing trace, MSE = 130.7676279320348

(a) Good prediction

(b) Good prediction

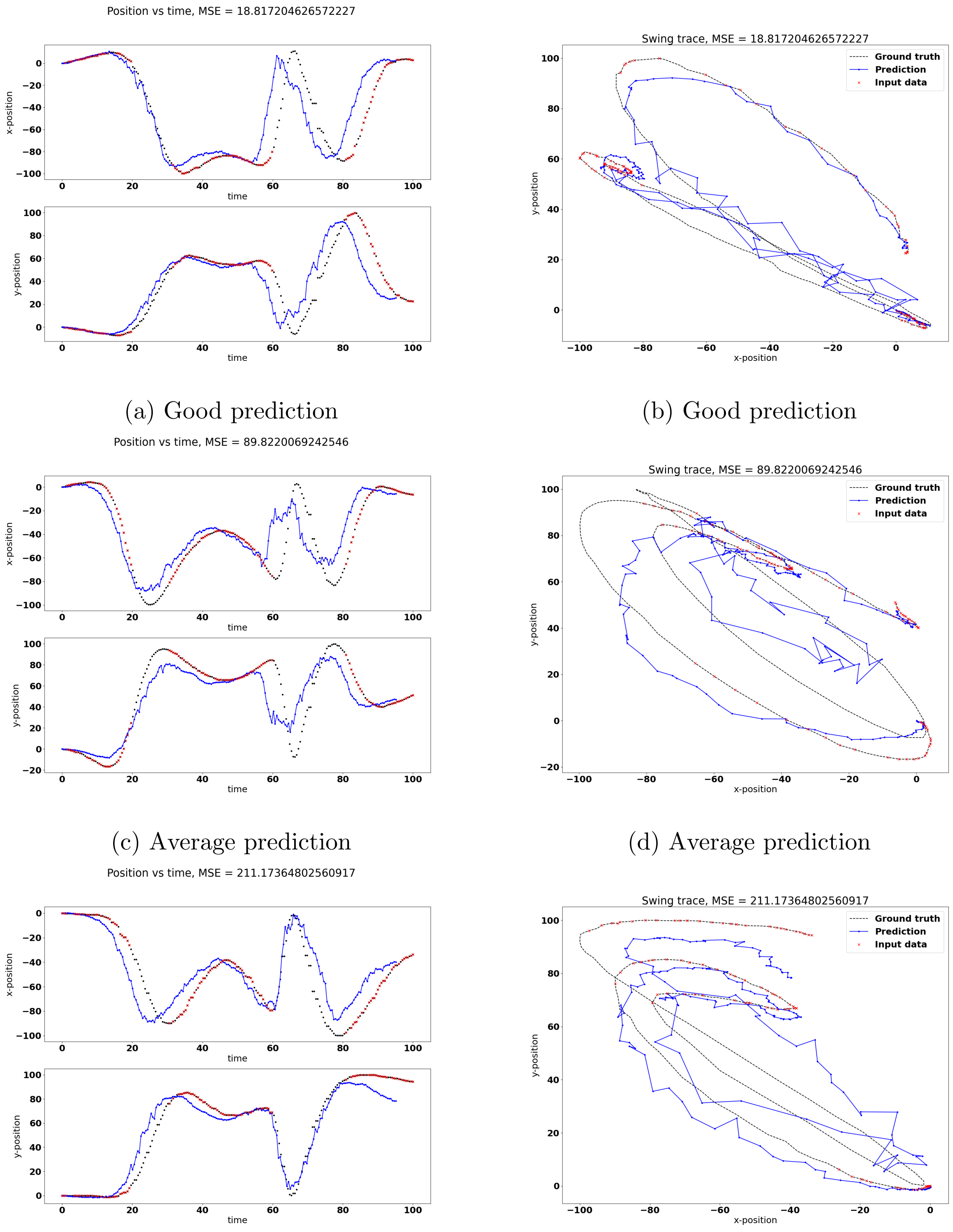
(c) Average prediction

(d) Average prediction

(e) Poor prediction

(f) Poor prediction

Figure 4.1: Example predictions sample size 125



(a) Good prediction

(b) Good prediction

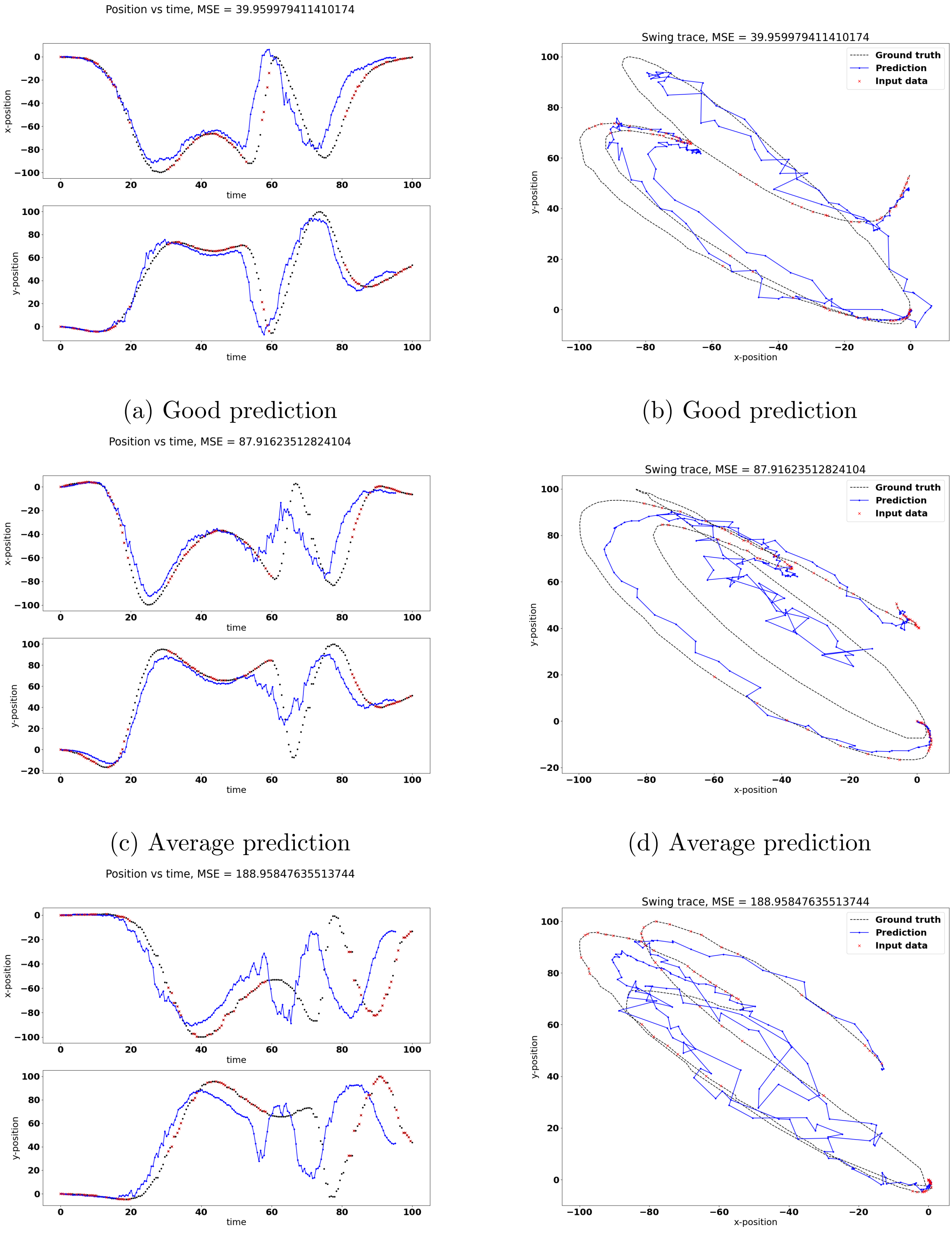
(c) Average prediction

(d) Average prediction

(e) Poor prediction

(f) Poor prediction

Figure 4.2: Example predictions sample size 100



(a) Good prediction

(b) Good prediction

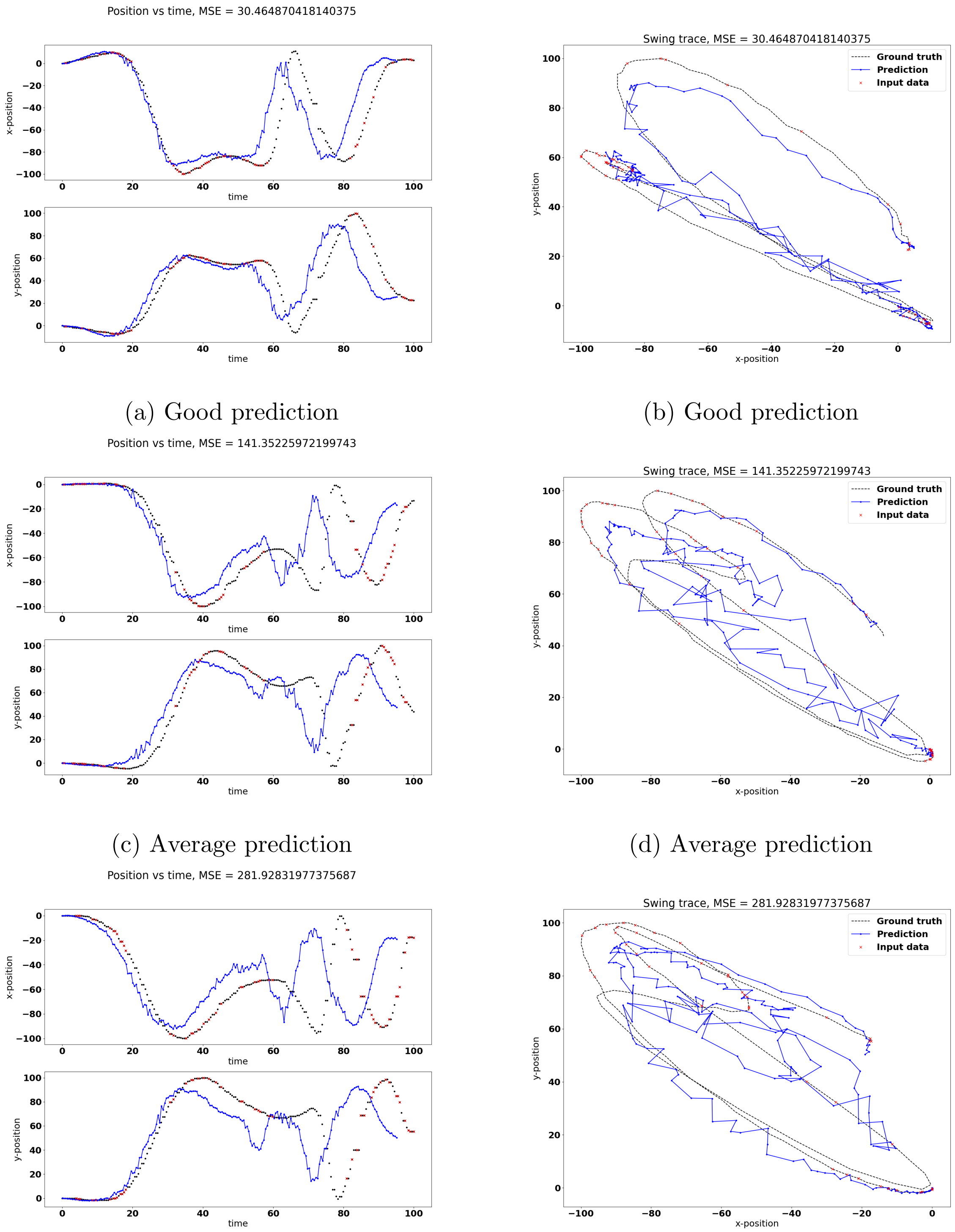
(c) Average prediction

(d) Average prediction

(e) Poor prediction

(f) Poor prediction

Figure 4.3: Example predictions sample size 75


Position vs time, MSE = 30.464870418140375

Swing trace, MSE = 30.464870418140375

(a) Good prediction

(b) Good prediction

Position vs time, MSE = 141.35225972199743

Swing trace, MSE = 141.35225972199743

(c) Average prediction

(d) Average prediction

Position vs time, MSE = 281.92831977375687

Swing trace, MSE = 281.92831977375687

(e) Poor prediction

(f) Poor prediction

Figure 4.4: Example predictions sample size 50

The average MSE of the NNs predictions on testdata for different sample sizes is presented in Table 4.2. As with XGBoost, the less information the network receives the higher is the MSE. For the NN however, the increase in MSE is steeper than for XGBoost as the sample size increases. The NN outperforms XGBoost in terms of MSE for sample sizes

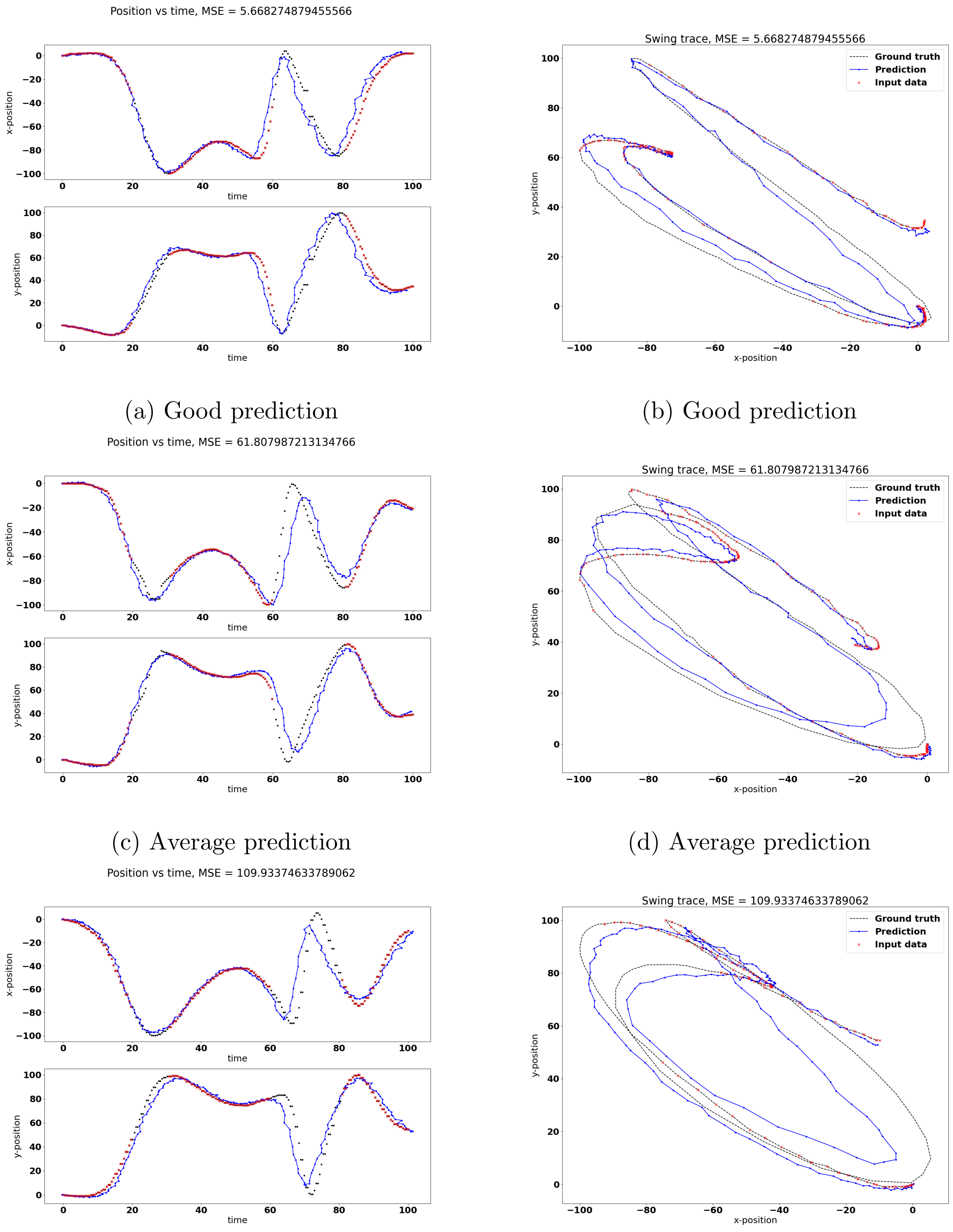
125 and 100, is about the same for 75, and has a higher MSE for 50.

Table 4.2: Average MSE of testdata predictions by the NN for different sample sizes

Sample size	MSE
125	51.10
100	64.73
75	89.12
50	155.08

Looking at the graphs on the left hand side in Figures 4.5, 4.6, 4.7, and 4.8, it is clear that the NN has learnt the underlying pattern of the club head's movement in the golf swing as well. For the most part, there is actually a very good match between the predictions and the ground truth. Albeit, just as with XGBoost, the match breaks down slightly for the smaller sample sizes.

The most significant difference between XGBoost and the NN becomes apparent when studying the graphs on the right hand side, which depict the traces. The predictions made by the NN have considerably less noise and look much more like actual traces.



(a) Good prediction

(b) Good prediction

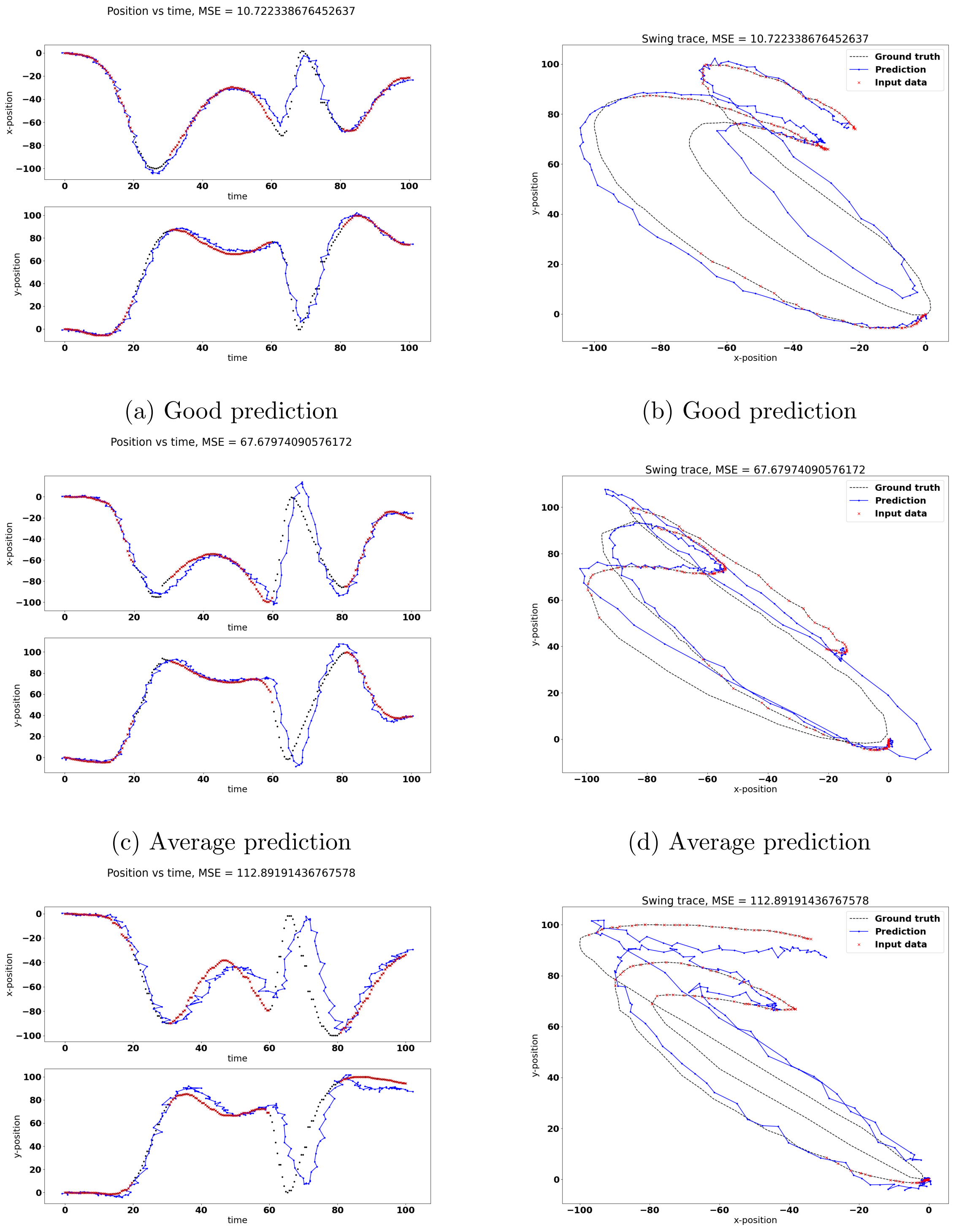
(c) Average prediction

(d) Average prediction

(e) Poor prediction

(f) Poor prediction

Figure 4.5: Example predictions sample size 125



(a) Good prediction

(b) Good prediction

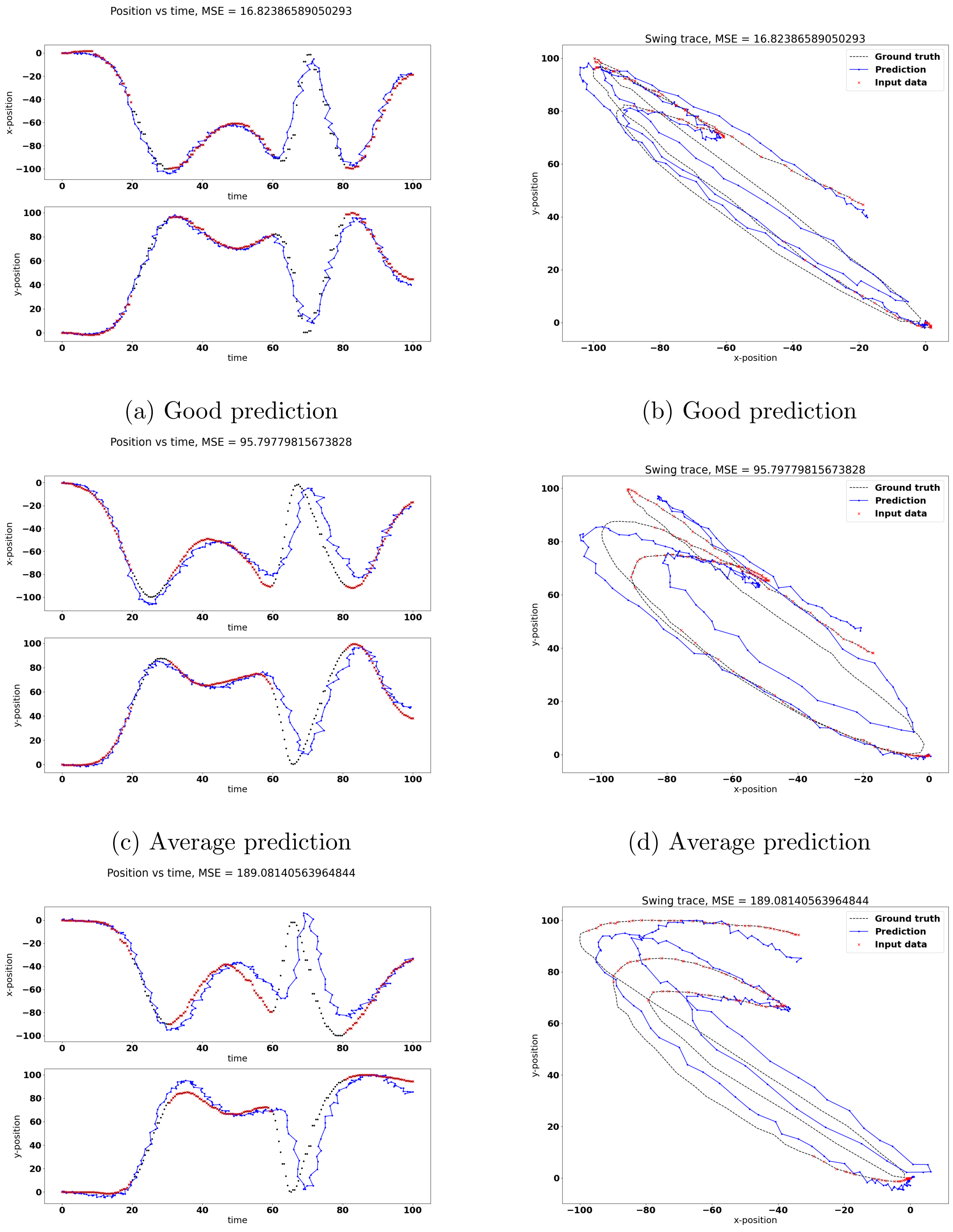
(c) Average prediction

(d) Average prediction

(e) Poor prediction

(f) Poor prediction

Figure 4.6: Example predictions sample size 100



Position vs time, MSE = 16.82386589050293

Swing trace, MSE = 16.82386589050293

Position vs time, MSE = 95.79779815673828

Swing trace, MSE = 95.79779815673828

Position vs time, MSE = 189.08140563964844

Swing trace, MSE = 189.08140563964844

(a) Good prediction

(b) Good prediction

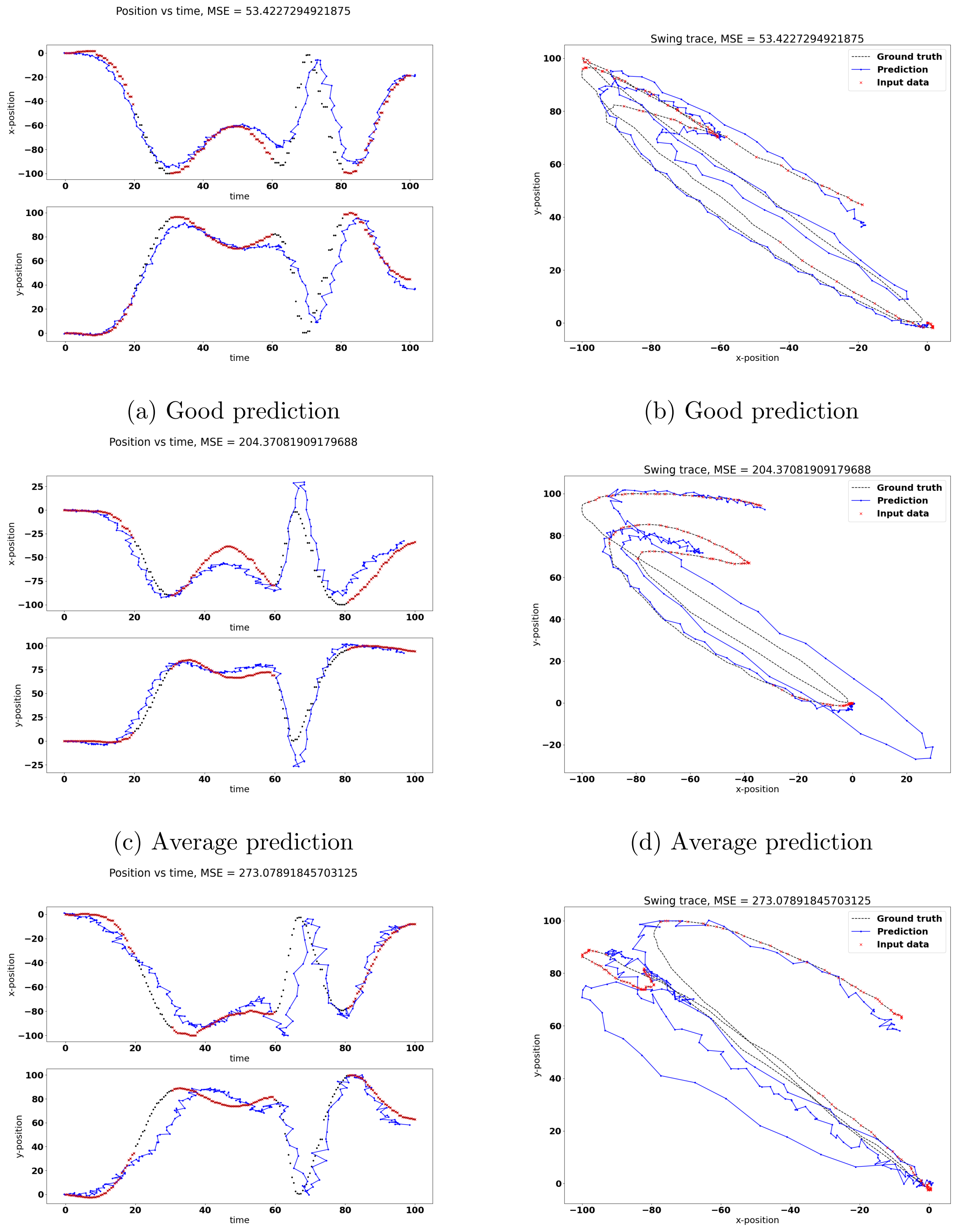
(c) Average prediction

(d) Average prediction

(e) Poor prediction

(f) Poor prediction

Figure 4.7: Example predictions sample size 75



(a) Good prediction

(b) Good prediction

(c) Average prediction

(d) Average prediction

(e) Poor prediction

(f) Poor prediction

Figure 4.8: Example predictions sample size 50

A commonality between the two models is that the part of the swing that is most challenging to predict correctly is the union of the latter part of the downswing and the beginning of the follow through. By design the models receive no input data in this region since no detections of the club head are expected when it is moving at such high speeds.

4.2 Size of the training dataset

The number of swings in the training dataset is altered to investigate the effect it has on the accuracy of the models' predictions. The size and contents of the test and validation datasets are kept unchanged. Table 4.3 shows the MSE of the testdata predictions of both models for different sizes of the training dataset.

The NN outperforms XGBoost for all the tested training dataset sizes.

Both models are improving their predictions as the size of the training dataset increases. The change in MSE is also quite large each time the training dataset is doubled. From these results it seems likely that the models would benefit from doubling the dataset since the MSE has not yet converged for either model. Each time the size of the dataset is doubled, it requires an ever larger time investment. Diminishing returns are expected on these investments, and can already be observed in Table 4.3. At some point the time investment required to double the dataset will be infeasible for the small increase in predictive power.

Table 4.3: MSE for different training dataset sizes, sample size 100

dataset size	XGB	NN
204	84.23	64.73
100	111.21	85.79
50	136.74	132.47
25	226.32	155.52
12	505.50	206.04

4.3 Augmentations

The effectiveness of augmenting the training data was tested for different sizes of the training dataset on the NN. Applying the rescale augmentation adds two augmented swings for each original swings. Applying the Squeeze/Expand augmentation adds four augmented swings for each original swing. Applying the Random Perturbations adds one augmented swing for each swing in the dataset, including augmented swings.

As can be seen in Table 4.4, the augmentations generally improve performance when the dataset is very small. However, for the larger datasets the augmentations only seem to make the predictions worse. The Rescale and Contract/Expand augmentations seem to perform better on the small datasets while the perturbations seem to be better for the larger datasets. In this test, combining multiple augmentations is never the best choice for any dataset size.

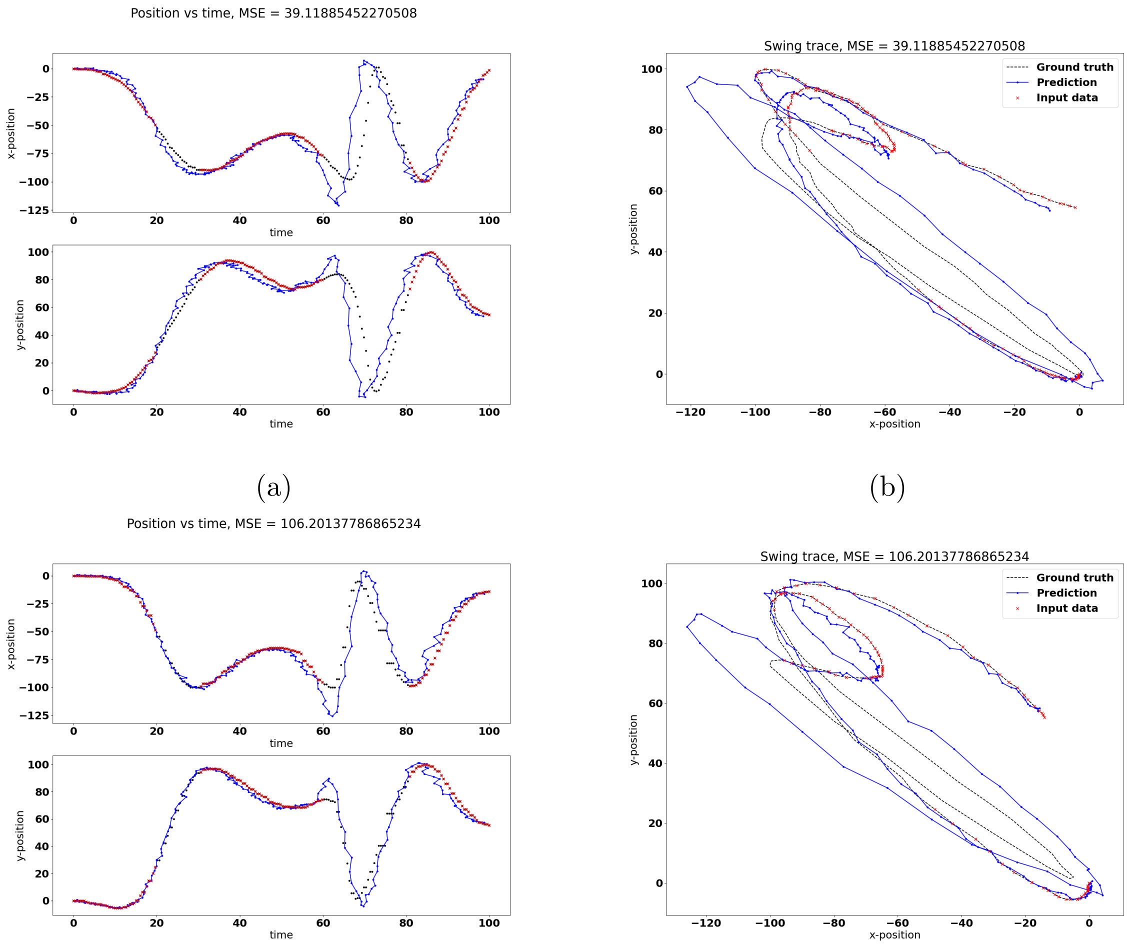
Table 4.4: MSE of the NN with sample size 100 for different Augmentations on varying dataset sizes

	25	50	100	204
No Augmentation	155.52	132.47	85.79	64.73
Rescale	145.64	125.60	97.49	80.66
Conract/Expand	128.16	132.03	94.77	90.48
Rescale + Contract/Expand	145.52	153.27	106.71	88.21
Small perturbations	155.97	135.92	77.28	70.45
All Augmentations	140.01	137.32	109.61	88.99

4.4 Limitations of MSE

In the graphs presented in subsection 4.1, it is evident that there is a high correlation between a low MSE and a good looking trace (this is especially true for the NN). However, MSE is still far from an ideal measure of how nice looking a predicted trace is. In the case of XGBoost, it becomes apparent that MSE only considers the distance from a predicted datapoint to the ground truth datapoint. This means that MSE does not care if the predicted datapoints are on alternating sides of the ground truth trace for each time step, creating a jagged and noisy trace, or if they are all on the same side creating a slightly off set but smooth looking trace. As long as the distances between the predicted datapoints and their respective ground truth datapoints are the same the MSE will be the same.

Furthermore, some additional examples of the limitations of MSE will be highlighted. The following predictions are generated by the NN with sample size 100 and 204 dapaoints in the training dataset. Figure 4.9 shows two similar predictions where both traces slightly overshoot in the beginning of the downswing. Despite the predictions looking virtually the same, one has a much lower MSE.

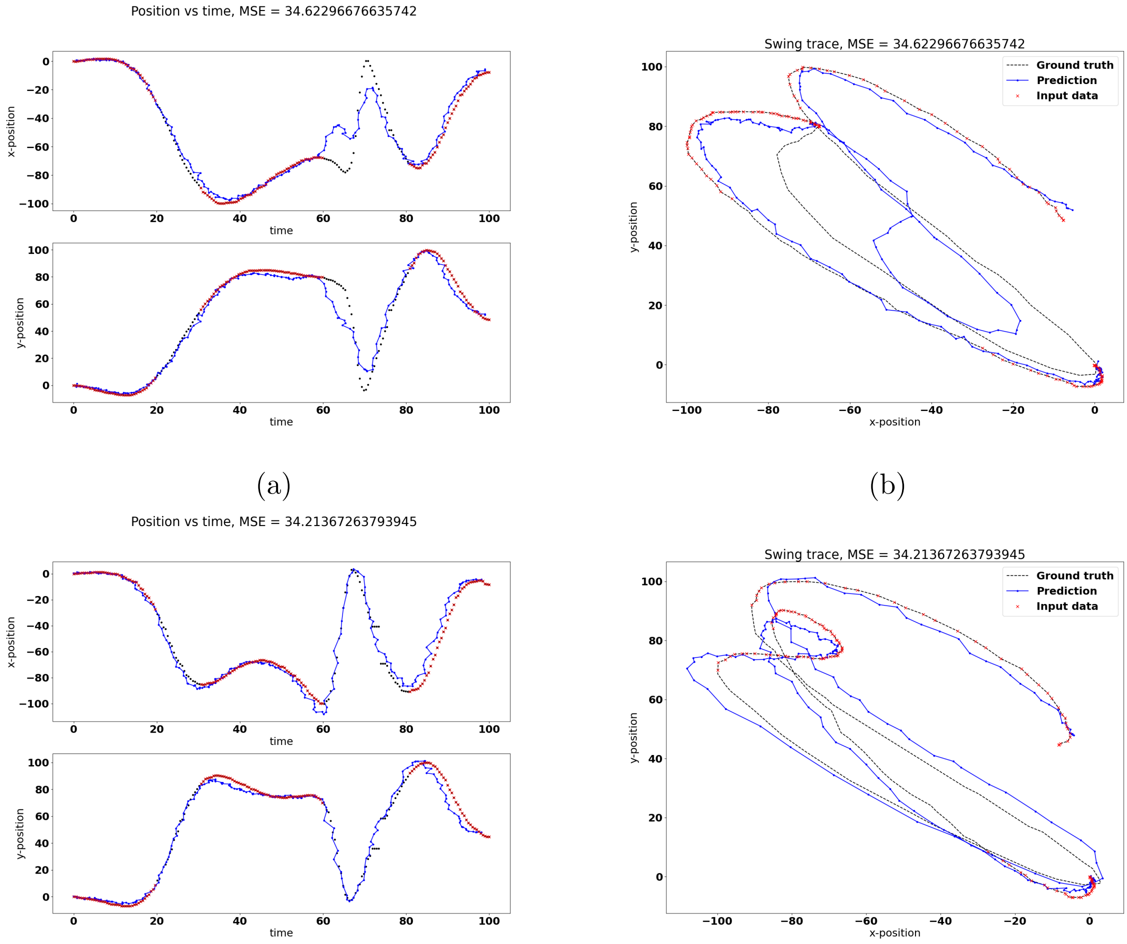


(a) (b)

(c) (d)

Figure 4.9: Example of two similar looking predictions with different MSE

Figure 4.10 shows two predictions with almost the same MSE. In spite of this, the prediction in (c) and (d) matches the shape of the ground truth quite well while the prediction in (a) and (b) makes a severe error in the downswing. Due to which the resulting trace barely resembles a golf swing. The fact that these two prediction almost have the same MSE could be explained by one being a very close match in all parts of the swing except the downswing, where the majority of the error occurs, whereas the other prediction is a little bit off in all parts of the swing and thus maintains a general shape that closely resembles the ground truth.

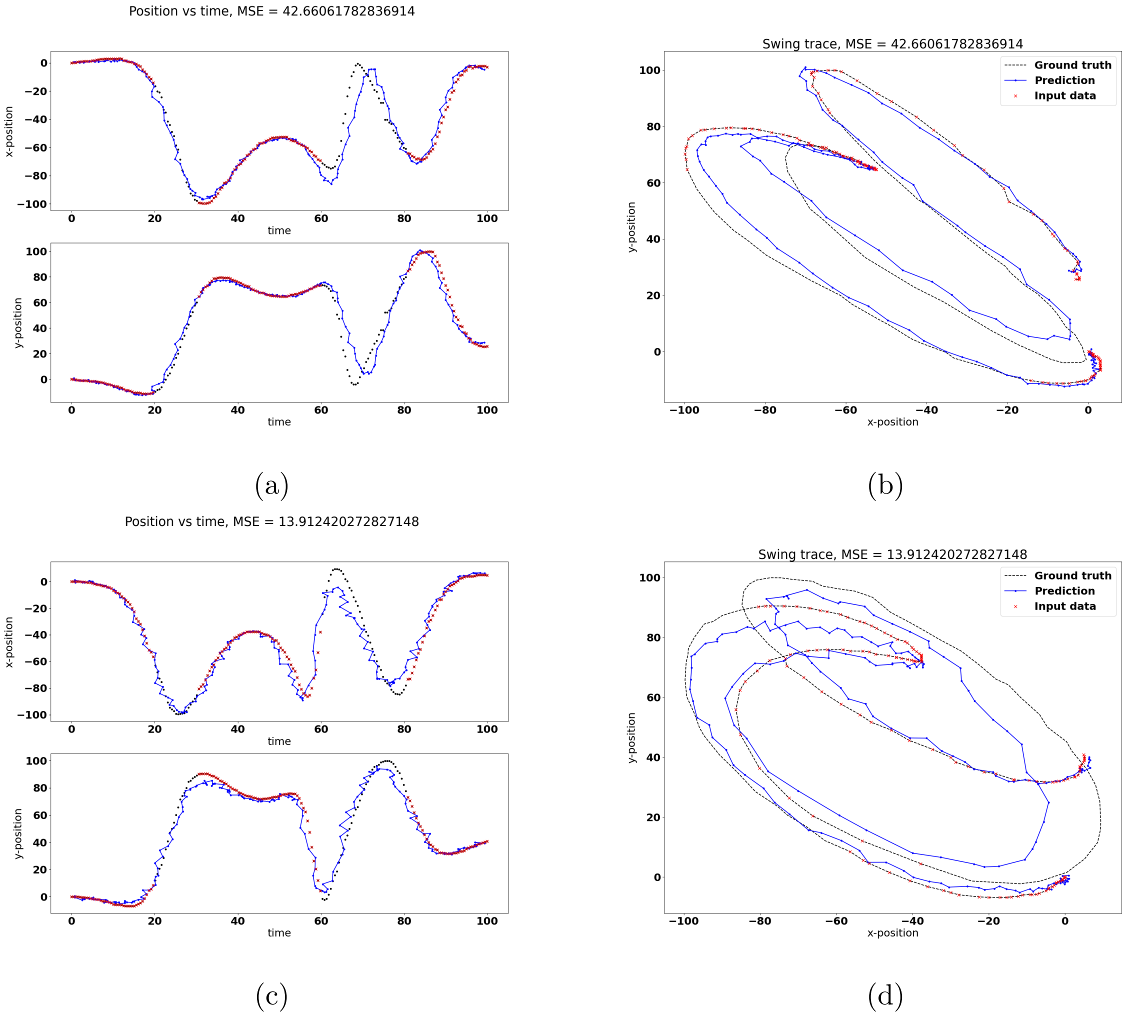


(a) (b)

(c) (d)

Figure 4.10: Example of two predictions with similar MSE where one looks better

Finally, Figure 4.11 presents two predictions where the trace of the one with the higher MSE arguably looks nicer. The reason why the prediction in (a) and (b) looks nicer than in (c) and (d) could simply be that the trace is smoother. As discussed previously, the smoothness of a trace, or lack thereof, does not impact the MSE.


Position vs time, MSE = 42.66061782836914

Swing trace, MSE = 42.66061782836914

Position vs time, MSE = 13.912420272827148

Swing trace, MSE = 13.912420272827148

(a)

(b)

(c)

(d)

Figure 4.11: Example where the trace with the higher MSE arguably looks better

5 Conclusions and Future Work

5.1 Conclusions

In this thesis machine learning was used to produce visual representations of golf swings, in the form of a trace, based on sparse detections of the club head in certain parts of the swing.

The collection and annotation of slow motion videos to generate training data proved to be one of the most time consuming parts of the project.

Two models were implemented, XGBoost and a neural network. Both models successfully learned the underlying patterns of how the club head moves throughout the golf swing. At least when they received an adequate amount of club head detections as input.

When it comes to producing traces that look realistic, the neural network performed better than XGBoost. In general, the traces produced by the neural network were much smoother and thus better matched the general shape of the ground truth trace.

In order for the models to be used in a live broadcast scenario, a lot more training data is needed to train the models and the models need to be improved upon further. Nonetheless, this thesis acts as a proof of concept showing that machine learning can be used to successfully interpolate club head detections to visually represent the golf swing as a trace.

5.2 Future work

This thesis shows promising results, but the models would need to improve a lot in order to be used in a real broadcasting scenario. The computer vision model can be expected to deliver about 10 to 25 detections of the club head. Therefore the model would need to reliably predict traces while only receiving 10 to 25 points as input data. Suggestions of improvements which lie outside the scope of this project are presented below.

More Data

One of the more obvious ways to improve the predictive capabilities of the models would be to simply generate and annotate more data. This is not a difficult procedure but it is very time consuming. Currently 256 swings have been annotated and that is simply not enough data.

Cost Function

Using a more complete cost function than MSE, which for example punishes noise, could result in more realistic predictions. The limitations of using MSE as the cost function for this application were made apparent in this thesis. The goal is really to produce good looking traces and minimising the MSE did not always achieve this. A measure that better describes how good a trace looks is required.

Return to Ball Position

The initial position of the club head at address is defined as the origin. This location is virtually the same as the ball position. Assuming that the player successfully strikes the golf ball (which is a very reasonable assumption to make), the trace could be made to always return back to the ball position at some point in the swing. A common error is precisely that the predicted trace fails to return to the ball, either over- or undershooting. The downswing/beginning of the follow through is the most difficult part to predict accurately. Anchoring the trace to the ball position might reduce some of the uncertainty in that part of the swing, resulting in better predictions.

Computer Vision Model

A different approach could be to improve the computer vision model so that it can deliver more detections. If it is able to produce more than 10-25 detections that would reduce the demands on the trace predictor.

Smooth Out Noise

A way to make the traces look like real golf swings would be to smooth out any noise present in the predictions.

Drawing the Trace

Another requirement for use in production is that the trace should be overlayed on the video and be drawn out following the club head.

Real Time Predictions

Finally, taking this concept to the next level would involve developing a model that can make predictions in real time. In other words, for the model to start predicting and drawing the trace before the golf swing is completed, so that it can be used live in a broadcast.

Finally, taking this concept to the next level would involve developing a model that can make predictions in real time. Meaning that the model starts to predict and draw the trace before the golf swing is completed, so that it can be used live in a broadcast.

Appendix

A Additional Average Predictions

A.1 Average predictions XGBoost

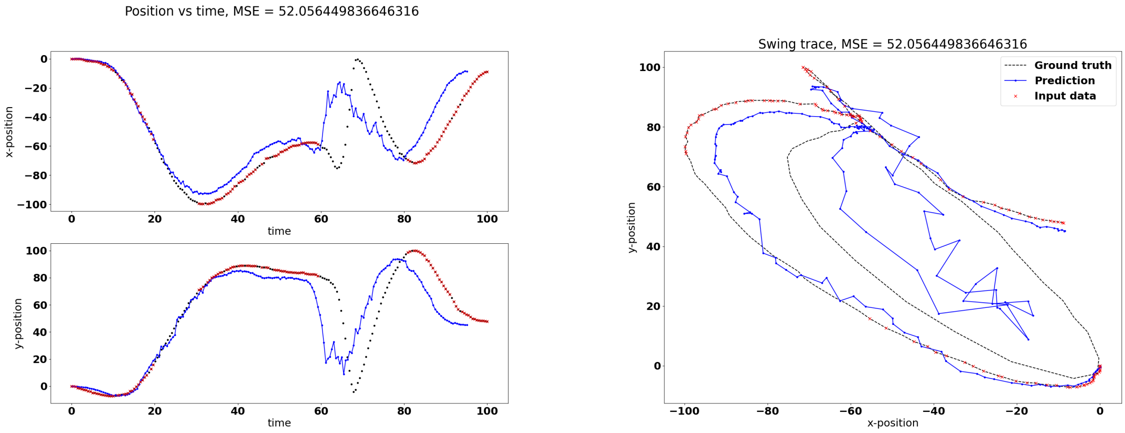
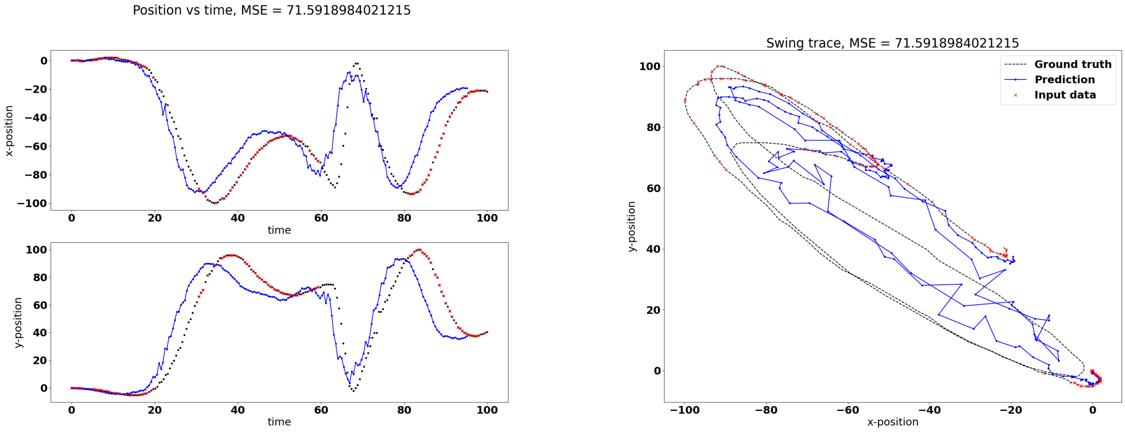


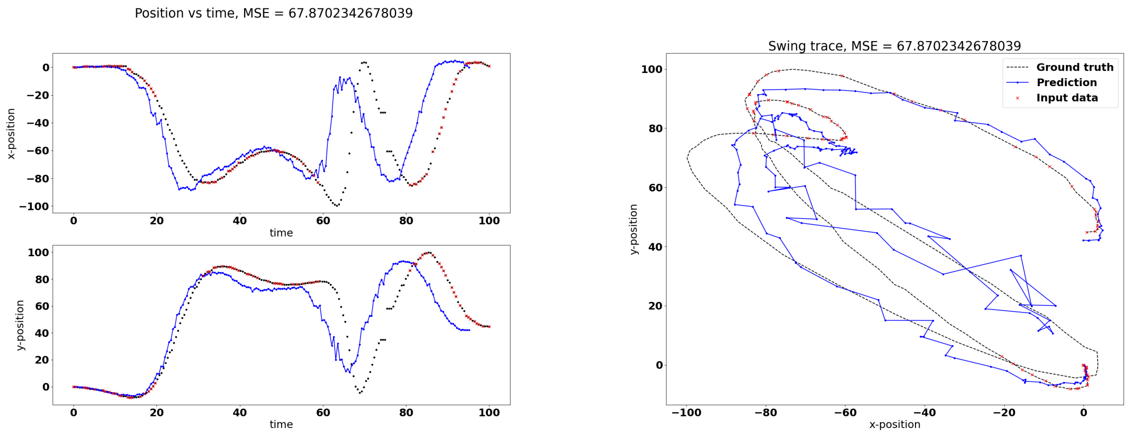
Figure A.1: Average prediction for sample size 125



(a) Average prediction

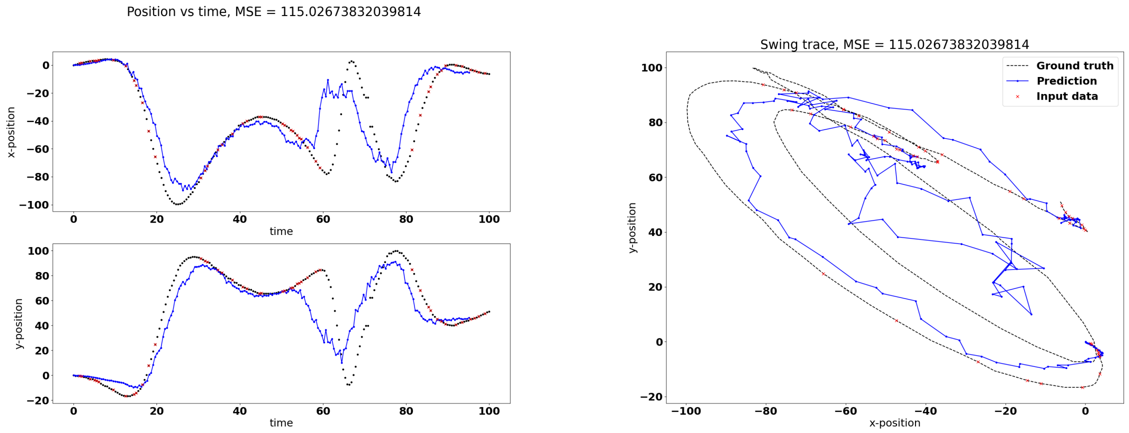
(b) Average prediction

Figure A.2: Average prediction for sample size 100



(a) Average prediction (b) Average prediction

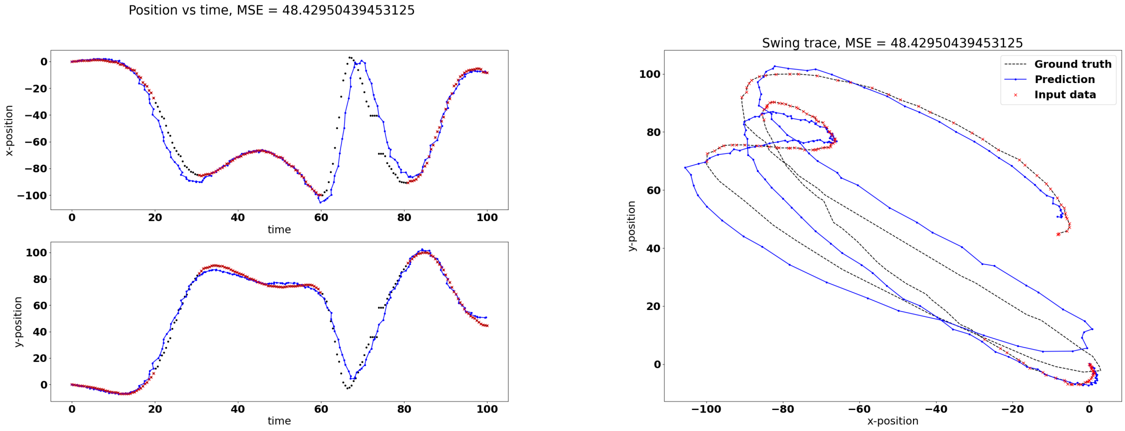
Figure A.3: Average prediction for sample size 75



(a) Average prediction (b) Average prediction

Figure A.4: Average prediction for sample size 50

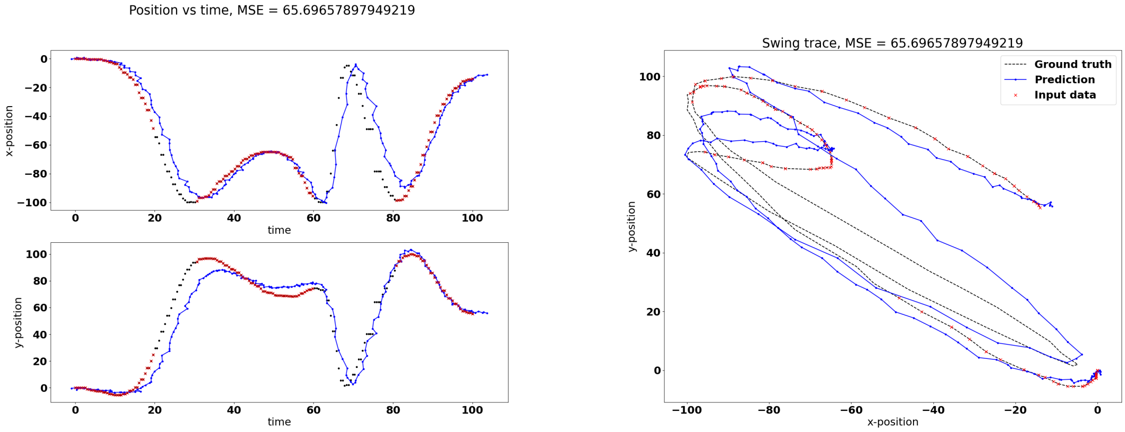
A.2 Average Predictions Neural Network



(a) Average prediction

(b) Average prediction

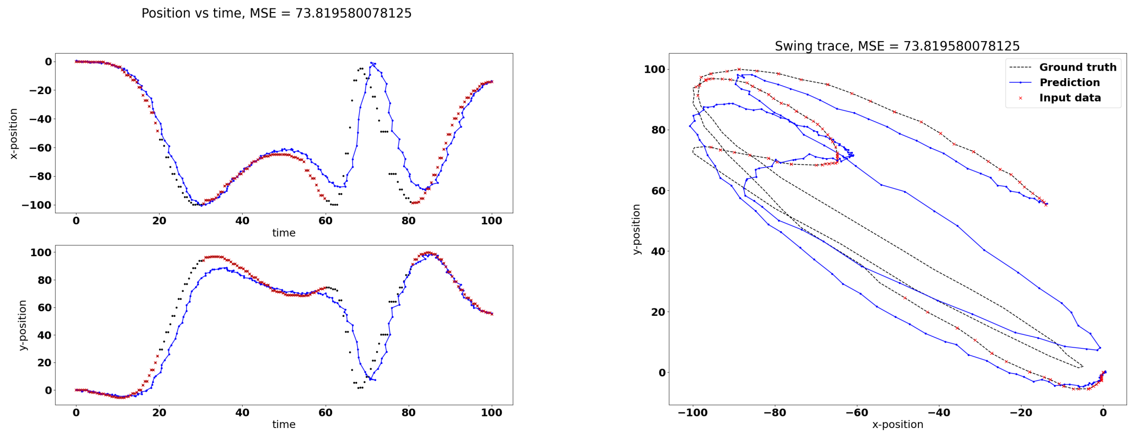
Figure A.5: Average prediction for sample size 125



(a) Average prediction

(b) Average prediction

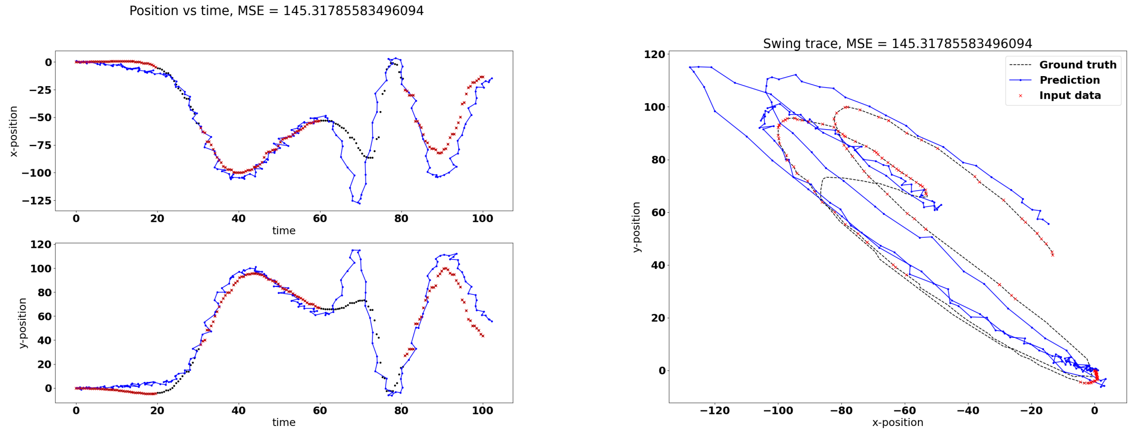
Figure A.6: Average prediction for sample size 100



(a) Average prediction

(b) Average prediction

Figure A.7: Average prediction for sample size 75



(a) Average prediction

(b) Average prediction

Figure A.8: Average prediction for sample size 50

References

[1] PGA TOUR. *Swing tracers from every player at the TOUR Championship.* [Accessed 7 July 2023]. 2021. URL: https://youtu.be/oZMv6SvpfvQ.

[2] PGA TOUR. *Rory McIlroy drives but they get increasingly longer.* [Accessed 7 July 2023]. 2022. URL: https://youtu.be/piuPCisIapc.

[3] TXG. *MATT IS BACK // TSR3 vs. TSi3 Drivers.* [Accessed 10 June 2023]. 2022. URL: https://youtu.be/c_zhEFo6Wt4.

[4] Golf Distillery. *Over the Top Golf Swing Error – Illustrated Guide.* [Accessed 10 June 2023]. URL: https://www.golfdistillery.com/swing-errors/over-the-top/.

[5] Chelsea Ortega. *Kinematics of the Golf Swing.* [Accessed 16 August 2023]. 2020. URL: https://www.evolutionphysicaltherapy.com/post/kinematics-of-the-golf-swing/.

[6] Greg Rose. *Kinematic Sequence Basics.* [Accessed 16 August 2023]. 2013. URL: https://www.mytpi.com/articles/biomechanics/kinematic_sequence_basics.

[7] Phil Cheetham. *The Linear Kinematic Sequence.* [Accessed 16 August 2023]. 2014. URL: https://www.mytpi.com/articles/biomechanics/the_linear_kinematic_sequence.

[8] *XGBoost.* 2018. URL: https://www.kaggle.com/code/dansbecker/xgboost/notebook (visited on 07/07/2023).

[9] *Introduction to Boosted Trees.* 2022. URL: https://xgboost.readthedocs.io/en/stable/tutorials/model.html (visited on 10/06/2023).

[10] Tianqi Chen and Carlos Guestrin. 'XGBoost: A Scalable Tree Boosting System'. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* KDD '16. San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 785–794. ISBN: 9781450342322. DOI: 10.1145/2939672.2939785. URL: https://doi.org/10.1145/2939672.2939785.

[11] Kevin Gurney. *An introduction to neural networks.* CRC press, 1997.

[12] AD Dongare, RR Kharde, Amit D Kachare et al. 'Introduction to artificial neural network'. In: *International Journal of Engineering and Innovative Technology (IJEIT)* 2.1 (2012), pp. 189–194.

[13] GOLFPASS. *MY ROOTS: RORY MCILROY My Voice.* [Accessed 16 August 2023]. URL: https://www.golfpass.com/learn/golf-tips-from-the-pros/rory-mcilroy.

[14] *XGBoost Documentation.* 2022. URL: https://xgboost.readthedocs.io/en/stable/ (visited on 10/06/2023).

[15] James Bergstra, Daniel Yamins and David Cox. 'Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures'. In: *Proceedings of the 30th International Conference on Machine Learning.* Ed. by Sanjoy Dasgupta and David McAllester. Vol. 28. Proceedings of Machine Learning Research 1. Atlanta, Georgia, USA: PMLR, June 2013, pp. 115–123. URL: https://proceedings.mlr.press/v28/bergstra13.html.