CHALMERS
UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG

Predicting Cross-Country Skiing Techniques Using Machine Learning

Evaluating performances of Random Forest Classifier and Long Short-Term Memory Neural Network to predict freestyle cross-country skiing techniques

Master's thesis in Computer Science and Engineering

SAVYA SACHI GUPTA

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2021

Master's thesis 2021

Predicting Cross-Country Skiing Techniques Using Machine Learning

Evaluating performances of Random Forest Classifier and Long Short-Term Memory Neural Network to predict freestyle cross-country skiing techniques

SAVYA SACHI GUPTA

UNIVERSITY OF GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2021

Predicting Cross-Country Skiing Techniques Using Machine Learning
Evaluating performances of Random Forest Classifier and Long Short-Term Memory Neural Network to predict freestyle cross-country skiing techniques
SAVYA SACHI GUPTA

Supervisor: Dan Kuylenstierna, Microtechnology and Nanoscience
Examiner: Moa Johansson, Computer Science and Engineering

Master's Thesis 2021
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2021

Predicting Cross-Country Skiing Techniques Using Machine Learning
Evaluating performances of Random Forest Classifier and Long Short-Term Memory Neural Network to predict freestyle cross-country skiing techniques
SAVYA SACHI GUPTA
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Power meters are vital for a cyclist during training and competitions as a tool for evaluation and improvement. Expanding their application to other sports would help athletes train better during workouts and maximize performance. In cross-country skiing, power comes from the poles and highly depends on skiing technique used. Thus, in order to predict power, it is important to predict which technique is being used. This thesis focuses on evaluating two machine learning methods to predict techniques in cross-country skiing. Data is collected in collaboration with Skisens AB who provided the sensors and Ulricehamn's skidgymnasium who helped in collecting data and experimental planning. Data was collected on a treadmill and on roller skis in an outdoor setting to get a balanced set of data and evaluations. Random forests and LSTM networks were selected as the two methods for evaluation. 10 fold cross validation was performed on each model after hyperparameter tuning and the overall accuracy, balanced accuracy and MCC score were recorded. Random forests with a reduced feature set achieved an overall accuracy of 74.4%, while the accuracy of treadmill data and outdoor data was 89.8% and 85.8% respectively. The LSTM model achieved an overall accuracy of 86.2%, while the accuracy of treadmill data and outdoor data was 86.5% and 84.1% respectively.

Keywords: Cross country skiing, machine learning, random forest, neural network, force, analytics, data science, sports, gait analysis, skiing techniques, science, engineering.

Acknowledgements

I would like to express my deepest gratitude to my supervisor Dan Kuylenstierna and my examiner Moa Johansson for their constant motivation, guidance and support throughout my thesis. I am thankful to them for their availability for discussions and sharing ideas at various stages of progress, especially during these unprecedented times.

This project would not have been possible without the entire group of skiers from Ulricehamn's Skidgymnasium, where the data was collected. I am grateful to all of them for spending their time and effort in ensuring the highest quality of information is collected. I would also like to thank Skisens AB who are responsible for developing the ski-pole handles that were used in the thesis for collecting data.

Finally, I would like to thank my parents, my brother, and my friends in all corners of the world for their continuous encouragement and support throughout the duration of the thesis. They were always available and did not let physical separation get in the way, even though the pandemic kept us all from each other. Without them, it would not have been possible to overcome the challenges I faced.

The success of this thesis was dependent on all these groups coming together in a coordinated manner, and for helping me achieve this, I would like to thank everyone!

Savya Sachi Gupta, Gothenburg, September 2021

Contents

List of Figures

List of Tables

1
Introduction

1.1 Overview

Power, or work-rate, is probably the most fundamental key indicator for comparison of human physical performance. In contrast to other metrics, e.g., speed, it is a robust measure to compare performance under various environmental conditions, e.g., wind or slope. It can also be viewed as a measure of absolute intensity that can be used for capacity analysis and for comparing different individuals. In contrast, measures such as heart rate, only provide the relative performance for an individual. With the use of power meters in cycling, some of the benefits mentioned above have been realized during training and competitions as a tool for evaluation and improvement of performance. Expanding the application of power meters to other sports would help athletes train better during workouts and maximize performance.

Another sport that has physical requirements very similar to cycling is cross-country skiing. In contrast to cycling, cross-country skiing has propulsion from both the poles and skis, and the power that comes from these poles is highly dependent on the skiing technique used. Thus, in order to assess the full-body power, it is important to identify which skiing technique (or gear) is being used. Gait analysis is a widely used technique that enables the study of body movement under various conditions and help identify patterns that can explain specific behavior, postures, etc. [1, 2, 3]. In cross-country skiing, gait analysis of an individual has usually involved IMUs mounted on the body at various locations and processing that data for analysis [4]. However, having sensors in ski-pole handles would enable data collection whenever and wherever a person is skiing, without the need for a separate IMU device.

Skisens[1] is a start-up company from Chalmers, that has developed a ski pole handle comprising of integrated sensors including a strain-gauge based load cell for force measurements and a 9 axis IMU for detection of the pole's angle versus the sagittal plane. This information is vital to identify the rhythm, speed, force of a skier as they ski in multiple gears. The handle is capable of processing and transmitting the information wirelessly to a smart phone for further analysis, training and potentially for performance improvement. If we successfully could identify skiing techniques using only the sensors in the ski pole, this would eliminate the need for additional

[1] `https://skisens.se/en-produkten/`

IMUs and create new opportunities in athlete training as they will now be able to gather data beyond just training facilities.

Now, if a machine learning method can be developed that would continuously learn from this readily available information, it could be used to assist skiers to improve their techniques, analyze current performance or provide recommendations for elite skiers and non-skiers alike. In the longer term, such a model could improve iteratively, the more frequently a skier uses the information to train and calibrate for their style of skiing. All this information could be made available directly on a mobile phone via interactive visualizations.

1.2 Literature Survey

Micro-sensor units that contain a combination of accelerometer[2], gyroscope and GPS sensors are being widely used in sports for gait analysis [5, 6, 7]. Even in cross-country skiing, such a unit mounted on the upper-back has helped identify the different skiing techniques (Marsland et al. [8]). These sensors measured 6 features - linear acceleration along the x,y,z-axis and angular acceleration along the x,y,z-axis. Using all these features, it was possible to identify cyclical patterns for each technique. It was also possible to observe skiing patterns unique to a skier by analyzing these patterns. This study was carried out in uncontrolled conditions with respect to speed, slope etc. due to which it was not possible to study the effect of these factors on the skiing patterns. However, another study [9] adds to this by controlling speed and slope on a treadmill and using five accelerometers attached to the hip, both the poles and boots. They were able to identify differences between skiing techniques and that even though the patterns may vary across skiers, a skier was still able to reproduce their pattern of skiing across training sessions within the same day or after 4 months.

Machine learning methods could leverage data from such sensors to predict and analyze performance. Neural networks such as Long Short Term Memory Networks (LSTM) is a popular machine learning method used for prediction of sequential data by learning long-term dependencies. LSTMs have been used previously to analyze cross country skiing techniques (Jang et al. [10]), where they used up to 17 sensors mounted on the body. Multiple models were trained using data from different number of sensors at specific positions on the body. Their study was performed on data collected by 3 skiers (1 male, 2 female) outdoors on flat and natural surfaces. They used a CNN-LSTM based deep learning model to test the different sensor configurations. It was observed that as the number of sensors reduced, the accuracy declined, but 5 sensors was the ideal balance between accuracy and lesser sensors. These 5 sensors were in similar positions as the previous work [9]. Here, the model performance was dependent on the location of the sensors, resulting in low accuracy even when a lot of sensors were used but located at inefficient positions. They

[2] https://www.analog.com/media/en/technical-documentation/application-notes/AN-1057.pdf

achieved a mean accuracy of 80% when trained over two skiers and tested over the third skier considered as unseen data.

Another study [11] made use of a single accelerometer inside a mobile phone that was mounted on the skiers chest. Data was collected for 11 skiers (7 male, 4 female) and a Markov-chain based machine learning model was employed that achieved an accuracy of 86.0% ± 8.9%. Here, it was found that the majority of the incorrect classification occurred while switching between gears. Some studies have tried to improve accuracy of a model by separately labeling these phases of time where a skier is switching between two skiing techniques so that they can be predicted as separate classes [12]. In this study, the classical style of skiing was considered and 6 sensors collectively measured data on the sternum, lower back and wrists, and with sensors directly on the pole handle itself using velcro straps. The skiers performed low intensity and high intensity runs on an outdoor track that enabled them to utilize most of the techniques available. Recorded data was split into cycles of strokes and supplied as input to a classification algorithm that achieved an accuracy >90%.

A pilot study was also conducted with Skisens using a neural-network-based approach [13], with the help of an earlier version of their ski-pole handle to identify skiing techniques using CNN, LSTM and Bidirectional-LSTM models. In this study, two different experiments were carried out - one where the model was trained on a subset of data containing samples from all skiers and tested on an unseen subset; and a second experiment where the model was trained on data from two skiers and tested on unseen data from a third skier. In the first experiment, the LSTM model yielded 95% accuracy and it was able to identify techniques accurately for all the skiers. However, the second experiment experienced a drop in performance resulting in an overall accuracy of 78%. It was suggested that a larger dataset with more skiers could potentially improve these results and such a model could generalize more effectively, as a model generally drops in performance when dealing with unseen data.

1.3 Objective

The objective of this thesis is to evaluate the performance of two machine learning methods to identify cross-country skiing techniques - specifically, 3 classical cross-country skiing techniques. The results obtained in this study could set a base for further development of applications that assist in training casual and elite skiers. Through this study, we want to assess if it is possible to predict skiing techniques using only two force sensors embedded in the ski pole handles. Compared to previous studies that use up to 6 inputs from each sensor (linear acceleration along the x,y,z-axis and angular acceleration along the x,y,z-axis), could a machine learning model using just 2 force measurements predict skiing techniques as accurately? Additionally, since data is being collected both indoors and outdoors, we want to assess how well the machine learning models can generalize and predict skiing techniques performed indoors and outdoors. It is important to identify the right machine learning method that would work well with time-series data and identify nuances in techniques either from available information or from additional calculated fields. The calculated fields would be created by leveraging domain knowledge in order to extract useful features of a skiing stroke. This would allow us to evaluate whether such calculated features could help understand the data and predictions more effectively. Lastly, we explore the feasibility of filtering methods to refine and group prediction results in such a way that results visually match real-world skiing scenarios better.

2

Background

This chapter provides background information that would be useful to understand the experimental setup and results. It begins by introducing the basic concepts of cross-country skiing and the techniques that are a part of this thesis. It covers details about the sensors and the machine learning algorithms used in the experiments. The metrics used to evaluate the the performance of the machine learning models are also discussed.

2.1 Cross-Country Skiing

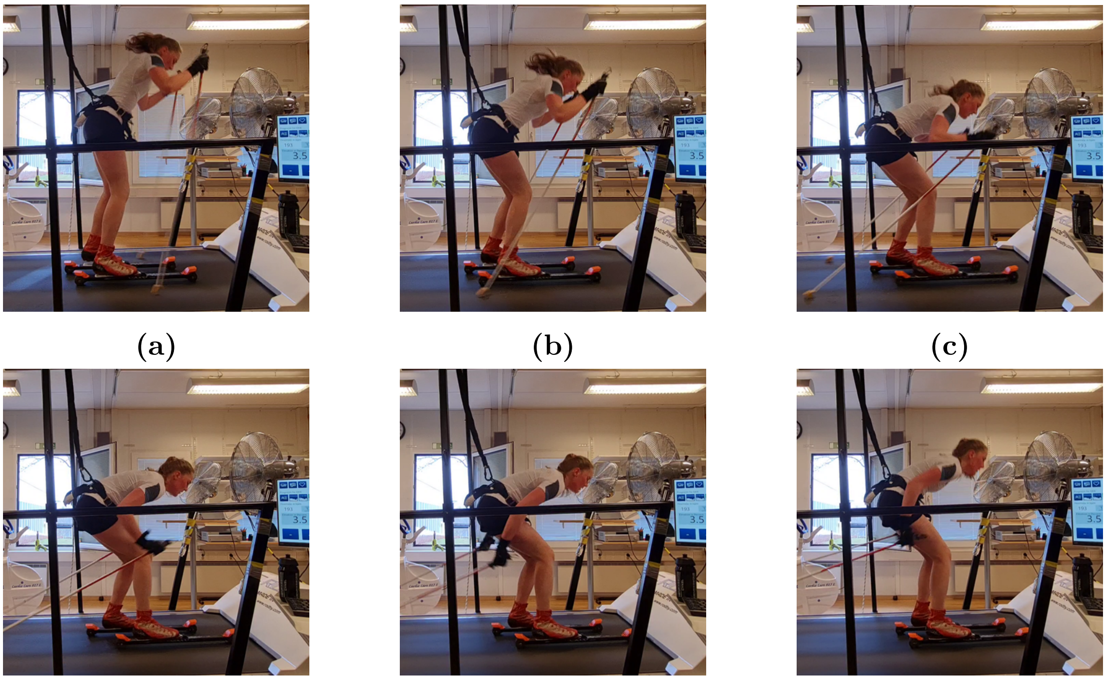
Cross-Country Skiing (abbreviated as xc-skiing) is a popular form of skiing, especially in the Scandinavian countries like Sweden, Finland and Norway. This form of skiing is typically done in landscapes comprising of flat terrains, gentle ascents and descents. Apart from competitive events, cross-country skiing is also popular among people who like to explore the outdoors whether they are seasoned professionals or beginners. Cross-country skiing requires basic equipment such as boots, skis and ski poles. Ski poles used in cross-country skiing are longer than usual as they aid in stability of the skier.

The skis are long and skinny, attaching to the boots only at the front part of the foot. This enables freedom of movement using the heels for specific maneuvers. This form of skiing requires the use of arms extensively as one tries to keep the body in motion and move forward specifically in ascents and flat terrains.

There are two main styles of cross-country skiing - Classical and Freestyle. This thesis focuses on identifying only the Classical techniques. For this thesis, three classical techniques were selected to analyze and identify using machine learning methods - Double Pole, Step Double Pole and Diagonal Stride.

2.1.1 Double Pole

The Double Pole (DP) technique is the most critical and commonly used technique in cross-country skiing. This technique can be broken down into three phases - Push-off, Glide and Weight Transfer. Fig. 2.1 depicts one stroke of the double pole technique. The stroke begins with the push-off phase where hands are held high, about a foot away from the shoulders with the elbows bent at a 90-degree angle (Fig. 2.1(a)). Once the skis are planted to the ground, the the upper and lower body is engaged to propel forward, by flexing the abs, knees and ankles (Fig. 2.1(b)-(f)). This is followed by the glide phase when the skier glides forward for a period of time that is dependent on factors such as terrain, wind conditions etc. As the skier is nearing the end of the gliding duration, they enter the weight transfer phase where they move up and forward by extending the upper and lower body that was flexed earlier.



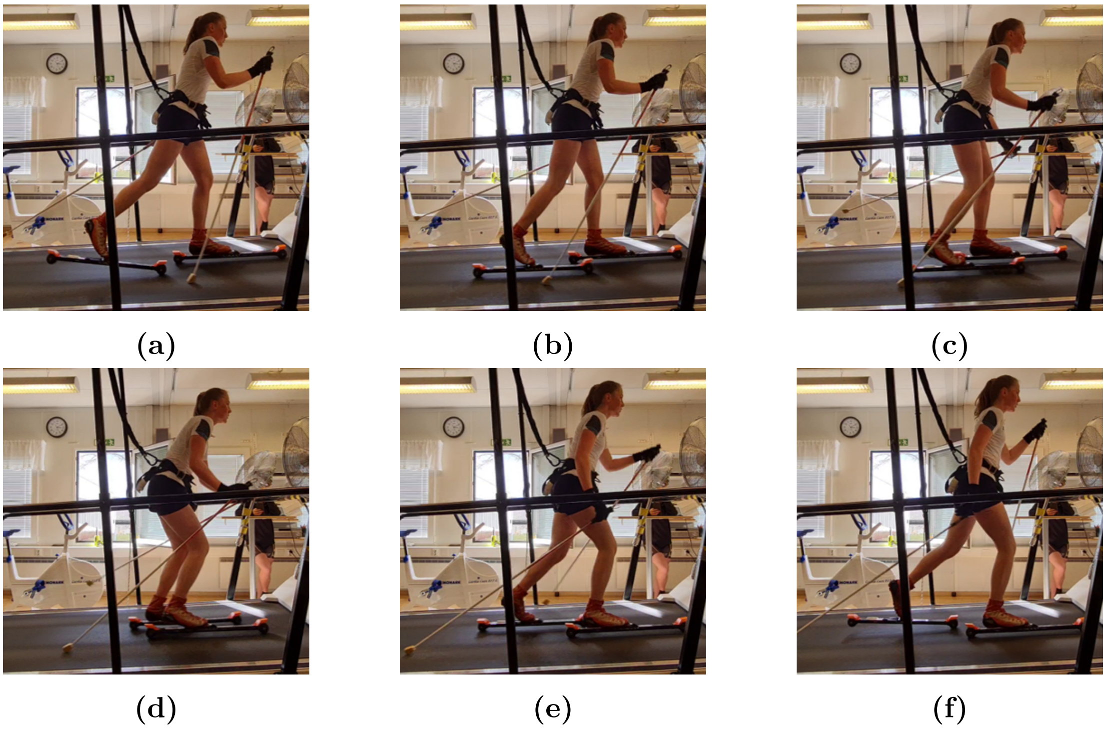
(a) **(b)** **(c)**

(d) **(e)** **(f)**

Figure 2.1: Double Pole Skiing Technique : Represented as a series of images going from (a) through (f)

2.1.2 Diagonal Stride

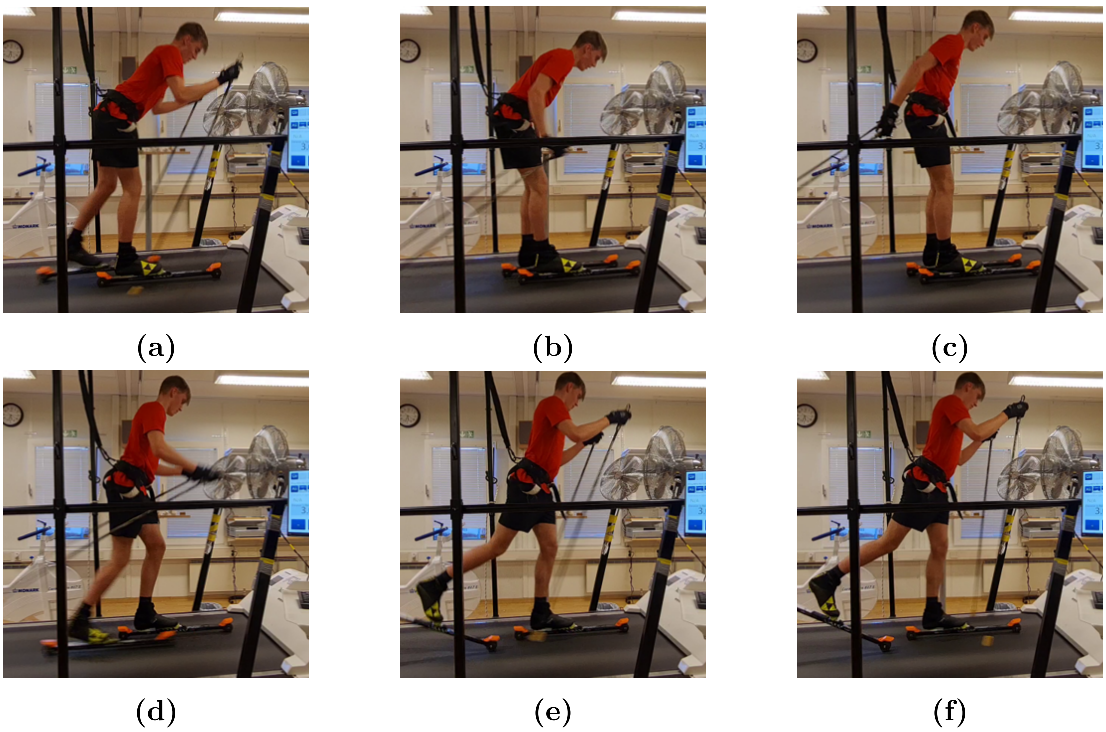
The Diagonal Stride (DS) is the oldest and most basic cross-country skiing techniques. It is a versatile technique characterized by fluid arm and leg movements that help power the skier across flats and uphills. Since these arm and leg movements are similar to walking, the diagonal stride serves as a good entry point into cross country skiing. The two basic elements of the diagonal stride are - a Kick, followed by a Weight Transfer. It begins by a downward facing back kick with right foot and swinging the right arm forwards and left arm backwards, while ensuring that the poles are angled backwards (Fig. 2.2(a)). The right pole must be planted in line with the left foot to facilitate gliding on the left ski (Fig. 2.2(b)-(c)). Then, the right foot must be brought back near the left foot to begin a kick with the left foot (Fig. 2.2(d)). This time, instead of the right foot, the left foot performs a kick while the left arm is swung forward enabling a glide on the right ski (Fig. 2.2(f)). The left foot is brought near the right foot and all the above steps are repeated again.



(a) (b) (c)

(d) (e) (f)

Figure 2.2: Diagonal Stride Skiing Technique : Represented as a series of images going from (a) through (f)



(a) (b) (c)

(d) (e) (f)

Figure 2.3: Step Double Pole Skiing Technique : Represented as a series of images going from (a) through (f)

2.1.3 Step Double Pole

The Step Double Pole (SDP) technique comprises of a double pole push and a single kick. It is a suitable middle ground between double pole and diagonal stride. A diagonal stride is a fast technique at high speeds with the risk of throwing the skier off balance, while the double pole may demand a lot of arm-strength that could get difficult for some skiers to keep up. The step double pole begins with a double pole push (2.3(a)-(c)) followed by a leg push as the skier is returning from the double pole (2.3(d)-(f)). The leg that is pushed backward recovers and returns to its original position as the skier prepares for the next double pole push. The kick is very prominent in this technique and needs to be timed accurately, in line with the forward swing of the arm of the double pole.

2.2 Power-Meters and Force Sensors

In sports such as cycling, devices called power-meters are used extensively. They are used to measure the power output of a rider and comprise of force sensors that deform upon application of force by a cyclist. Using this force measurement one can calculate the power (in Watts). Power meters are vital in enabling efficient training and performance improvement.

These force sensors consist of a load cell that typically measures a point force. In the ski handles provided by Skisens AB, the force sensor is embedded in the handle and it measures the force exerted by the skier during each pole push. The sensors have a sampling rate of 100 Hz and yield files that can be read by a MATLAB code/ dedicated mobile application. The ski pole handles provided by Skisens are shown in Fig. 2.4. Fig. 2.5 shows these sensors attached to the ski poles, ready to be used.

(a) Top view of the ski pole

(b) Side view of the ski pole handles

Figure 2.4: Ski pole handles with force sensors, provided by Skisens. 2.4a shows a button to activate and pair handle to the app.

Figure 2.5: Ski pole handles attached to ski poles

2.3 Machine Learning Methods

The machine learning methods selected for the thesis were - Random Forest Classifier and Long Short-Term Memory (LSTM) neural network. Random forests are known to be good classifiers and can work with high dimensional data. They can also be tuned to reduce over-fitting and can generalize well. Additionally, there has been limited use of random forests in other gait analysis applications [14, 15] and it would be interesting to see how random forests perform for this application. On the other hand, neural networks such as LSTM's, have been used earlier to predict skiing techniques with different types of experimental setups. However, most of the projects made use of multiple body mounted sensors or worked with limited amount of data. In this case, data from just two handle based sensors will be used in addition to collecting data across different locations which could result in different performance of the model and the generalization capability of the model can be assessed.

2.3.1 Random Forest Classifier

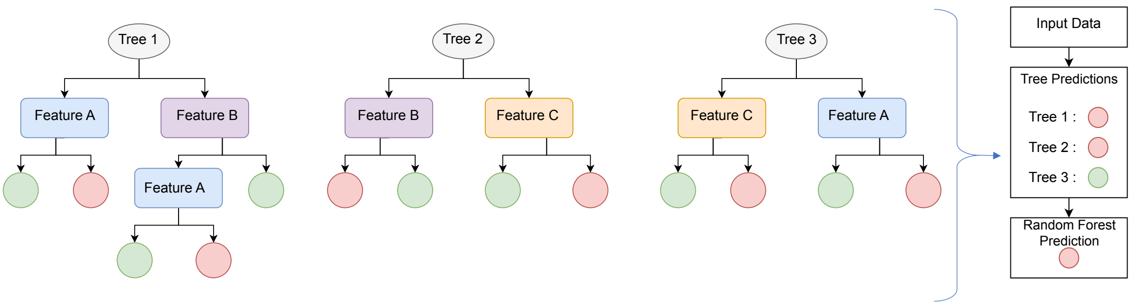


Figure 2.6: Example of a simple random forest classifier

Random forest classifiers are algorithms that build a group of 'Decision Trees' called 'Forests', and combine them together to create more stable and accurate predictions [16]. Decision trees resemble flowcharts such that each node represents a condition or test on a feature of the input dataset. Based on the result of the test, the node branches into further nodes. This is repeated at each node until no further splitting is possible a final node known as the terminal node is reached that contains the class label. Random forests contain numerous such decision trees where instead of using the most important feature to split a node, it selects the best feature from a random subset of features. This variability results in more accurate/stable predictions.

Random forest classifiers are also popular for providing an attribute called 'Feature Importance' that measures the relative importance of each feature on the prediction. using the Python programming language, this attribute is automatically calculated using inbuilt functions and scales the results so the sum of all importance is equal to one. Feature Importance can play an important role in deciding which features to include or exclude based on how much they impact the predictions. This can come handy to avoid the problem of over-fitting - where the model is tailored too much towards the training data.

2.3.2 Long Short-Term Memory (LSTM) Neural Network

Long short-term memory (LSTM) [17] networks are recurrent neural networks that are used for classification and prediction of time series data where the network may learn dependencies in sequential data. Recurrent neural networks (RNN), are used for analysis of time-dependent data and work well with short term dependencies. However, RNN's are unable to understand context and learn long term dependencies with back-propagation as the computations can cause the gradient to tend to zero or infinity. These are called the vanishing gradient and exploding gradient problems respectively. This is where an LSTM comes in, as it is capable of learning long-term dependencies. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.

2.4 Evaluation Metrics

In order to effectively quantify the performance of a machine learning model, evaluation metrics are used. They give an insight into the predictive capabilities of a model by computing various measures using the original/true labels and the predicted labels. There are different kinds of evaluation metrics and using multiple metrics can give a better estimate of the model performance as a model may perform well for one metric but not the other. This section describes the evaluation metrics that were selected for the purpose of this thesis.

Confusion Matrix

A confusion matrix is a widely used method to evaluate the performance of a classification model. It is an N x N matrix where N is the number of labels that are predicted. The true labels are depicted on one axis and the predicted labels on the other that assist in calculating other evaluation metrics. Fig. 2.7 depicts a simple confusion matrix for a binary classification problem. Fig. 2.8 represents confusion matrices for a multi-class classification problem where each sub-figure denotes the confusion matrix with respect to a specific label.

True Positive

This value represents the number of true values matching the predicted values for the positive label.

False Positive

This value represents the number of values that were predicted incorrectly, i.e., when the predicted value was positive but the true value was negative.

True Negative

This value also represents the number of true values matching the predicted values, but for the labels that are not used for calculation of true positive, i.e., the number of true negatives matching the number of predicted negatives.

False Negative

This value represents the number of values that were predicted incorrectly, i.e., when the predicted value was negative but the true value was positive.

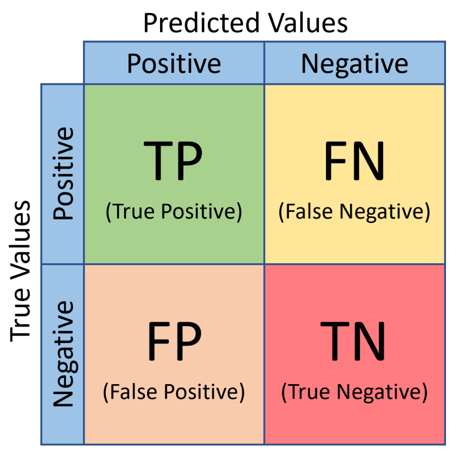
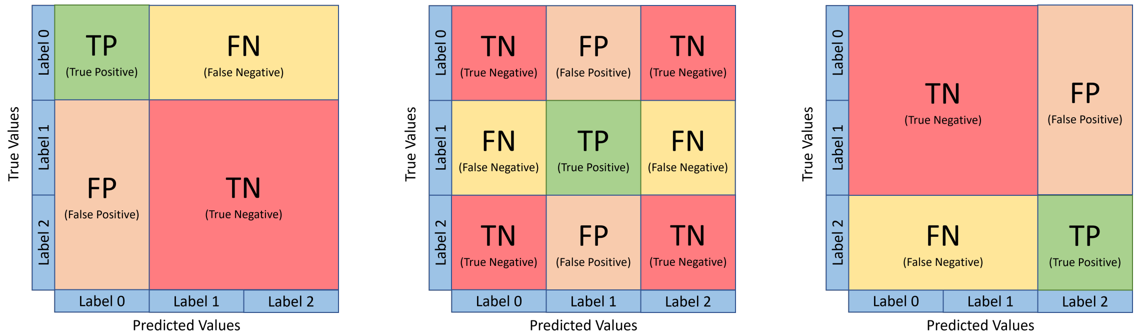


Figure 2.7: Simple confusion matrix for binary classification



(a) Confusion Matrix for Label 0

(b) Confusion Matrix for Label 1

(c) Confusion Matrix for Label 2

Figure 2.8: Confusion matrix for multi-class classification

Accuracy

Accuracy is the number of correct predictions made by the model out of the total predictions made. From the confusion matrix, the correct predictions are represented by the true positives and true negatives. The complete formula is given by Eq. 2.1. For a multi-class classification problem, the accuracy is given the sum of true positives of every class divided by the total number of predicted values. However, accuracy is highly sensitive to class imbalance. It does not take into account the disparity in the number of values in each class and the overall accuracy can be skewed towards any major class.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

Balanced Accuracy

Balanced accuracy takes class imbalance into account to give a better estimate of the models performance. It can be calculated as the mean of the accuracy of each of the classes calculated individually.

Matthews Correlation Coefficient

Matthews Correlation Coefficient (MCC) is an evaluation metric considered to be reliable and informative as it takes into account values from all four categories in the confusion matrix. Eq. 2.2 depicts the MCC formula for binary classification. It yields a high score only if the model performs well in all these categories. It also provides as added advantage of handling class imbalance well. Similar to correlation coefficients, this metric returns a value between -1 and 1 where a high value (close to 1) means that classes are predicted well and a low value (close to -1) implies a total disagreement between true value and prediction. An MCC score of 0 means that the classifier is no better than a random flip of a fair coin.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)\,(TP + FN)\,(TN + FP)\,(TN + FN)}} \tag{2.2}$$

3
Methods

This chapter describes the data and introduces the machine learning models used to carry out the experiments. It explains the process of collecting real world data from skiers and performing additional processing steps so that it is ready to be input to the machine learning models. The preliminary observations and patterns are discussed as part of the exploratory data analysis to assist in performing the experiments. Finally, both the machine learning models are introduced covering aspects such as - type of input data, hyperparameter tuning and model architecture.

3.1 Data Collection

Data collection was carried out in collaboration with Skisens who supplied the sensors and Ulricehamn's skidgymnasium that has a treadmill allowing skiing under controlled conditions. High-school athletes at Ulricehamn's skidgymnasium assisted with data collection and experimental planning. Additionally, video footage was also made available of the indoor treadmill runs in order to monitor consistency in the sensor readings and user actions. This helped in understanding if there was any drift in the sensor data and if there was a need to apply any correction for the same. For the purpose of the thesis, data was collected in the following scenarios:

1. **Indoors** : Controlled data collection based on pre-defined protocol in an indoor environment on a treadmill. The treadmill protocol defined the duration, speeds, slopes and techniques to be used by a skier while collecting data. The protocol was defined to be 23 minutes long, with a change in slope and/or speed every 30 seconds. At pre-defined times the skier would switch between techniques thereby ensuring that all techniques are covered. The detailed treadmill protocol is provided in Appendix. A.1.
2. **Outdoors** : Free-form data collection performed outdoors on roller-skis. In this scenario, the duration, speed, slope and techniques used was dependent on the skiing location. Given the variable geographical nature of the location, it is not possible to adhere to a pre-defined protocol. However, the skiers attempted to capture all techniques as best as possible. The techniques used while collecting this data was labeled by the skiers after each session.

Since the skiing techniques vary in intensity, it was not possible to get the same number of strokes for each technique. Thus, instead of attempting equal number of strokes for each technique, the dataset was balanced during the machine learning process.

A total of 14 skiers volunteered for data collection by collecting 21 sets of data for analysis. On the treadmill, 11 skiers - 6 Male and 5 Female collected data based on the treadmill protocol and 3 Male skiers collected data outdoors. Some of the skiers provided multiple data sets, such as the outdoor data collection where 3 skiers collected data across 3-4 sessions each. No skier collected data across both treadmill and outdoor sessions. Table 3.1 provides the details of skiers for each session. The height and weight distribution of the skiers is represented in Fig. 3.1.

Table 3.1: Skiers Profile for Data Collection

Ski Session	Skier Name	Gender	Mass (kg)	Height (cm)	Pole Length (cm)	Ski Location
1	Skier A	Male	90	190	160	Treadmill
2	Skier B	Male	75	190	160	Treadmill
3	Skier C	Male	75	175	145	Treadmill
4	Skier D	Male	75	175	145	Treadmill
5	Skier E	Female	71	179	149	Treadmill
6	Skier F	Female	56	158	140	Treadmill
7	Skier G	Male	86	187	157	Treadmill
8	Skier H	Male	77	177	149	Treadmill
9	Skier I	Female	60	172	148	Treadmill
10	Skier J	Female	60	173	148	Treadmill
11	Skier K	Female	49	156	135	Treadmill
12	Skier L	Male	65	175	147	Outdoor
13	Skier L	Male	65	175	147	Outdoor
14	Skier L	Male	65	175	147	Outdoor
15	Skier M	Male	73	180	150	Outdoor
16	Skier M	Male	73	180	150	Outdoor
17	Skier M	Male	73	180	150	Outdoor
18	Skier M	Male	73	180	150	Outdoor
19	Skier N	Male	80	180	150	Outdoor
20	Skier N	Male	80	180	150	Outdoor
21	Skier N	Male	80	180	150	Outdoor

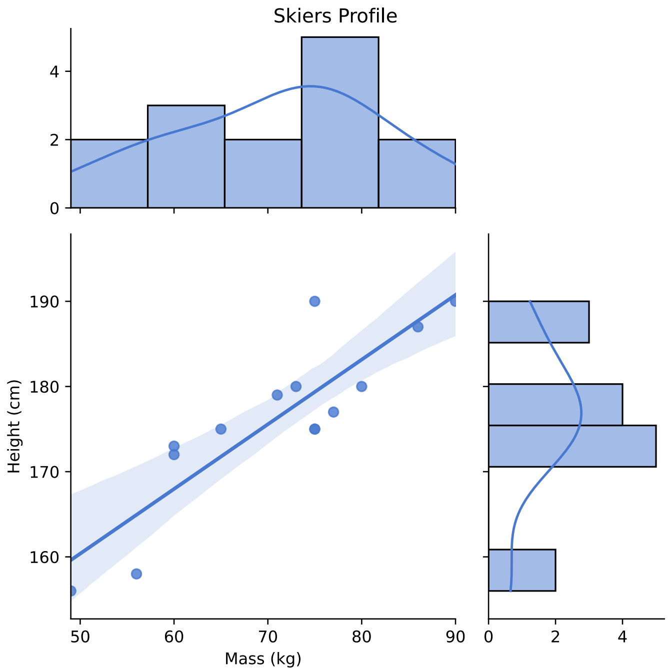


Figure 3.1: Height and weight distribution of skiers

3.2 Pre-Processing Data

The force data was stored in '.bin' files which were converted to '.csv' files using a pre-existing MATLAB program provided by Skisens so that the Python code could work with the data effectively. The '.csv' files comprised of the force and timestamp data along with other parameters such as slope, speed in cases where the data collection was carried out on a treadmill. For data collected outdoors, the columns for slope and speed were left blank as that data was not available. For the purpose of this thesis, only the force data is of interest. Table 3.2 highlights the fields available for analysis. The '.csv' fields are further detailed in Appendix. A.2.
In addition, additional calculated fields were created - identifying the time interval between strokes from the left and right hand, frequency of the strokes, duration of the stroke when the pole was in contact with the ground and/or air, etc. These additional features would be useful while evaluating non-neural network methods such as Random Forest Classifiers. Depending on the method we choose for building each of the models, we could either use a time-series data directly, or input feature properties derived from the time-series to non-neural network models.

Table 3.2: Data collected for analysis

Measure	Description
Time Left (min)	Timestamp of the left ski pole
Time Right (min)	Timestamp of the right ski pole
Force Left (N)	Force of the left ski pole

Force Right (N)	Force of the right ski pole
Slope (%)	Slope angle of surface
Speed (km/h)	Speed of Skier
Technique	Technique used by Skier

3.2.1 Data Cleaning

Upon plotting the force values over time, it was observed that there were long periods of inactivity recorded in the poles. This was more prevalent with the outdoor datasets where it was more likely that a skier would not be using the poles at certain intervals due to the skiers own momentum, or while going down a gradual slope. However, these inactive intervals were labeled as 'Double Pole'. This mislabeled data would have affected the prediction performance of the model as it does not belong to any particular technique and was thus removed. This resulted in a more compact dataset comprising of force data only from the techniques performed by the skiers. The neural network models used this cleaned data for identifying the skiing techniques.

3.2.2 Creating Calculated Features

The sensors had a sampling rate of 100Hz, i.e., capturing 100 force values every second which resulted in an extremely large dataset. Random forests are known to not scale well to large datasets making it difficult to use forests on our dataset in its current form. Thus, calculated features that give information about multiple aspects of a stroke were created using the force data, thereby reducing the size of the dataset and creating meaningful features. This also provided the opportunity to compare the different skiing techniques based on the calculated features and understand if there are major differences in the way these techniques are performed.

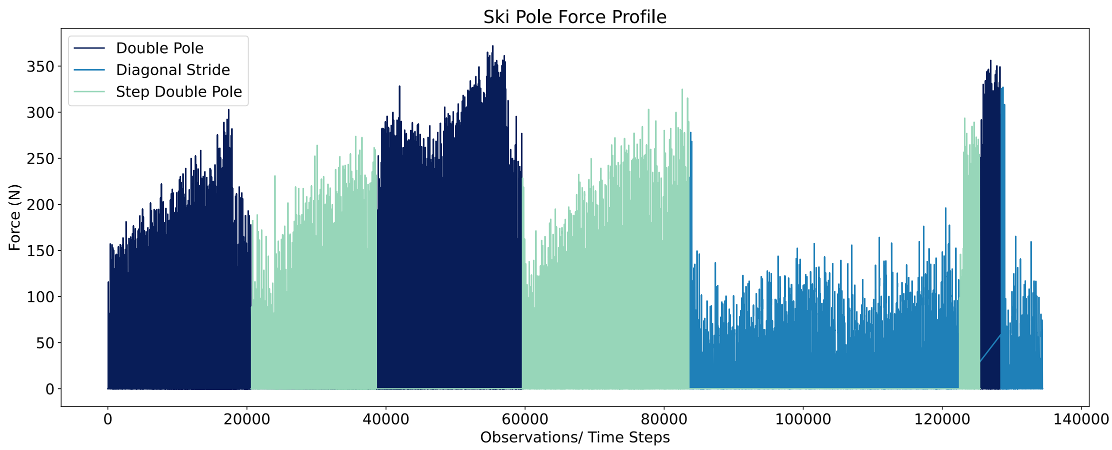


Figure 3.2: Sample force profile from the left ski pole of a skier on a treadmill

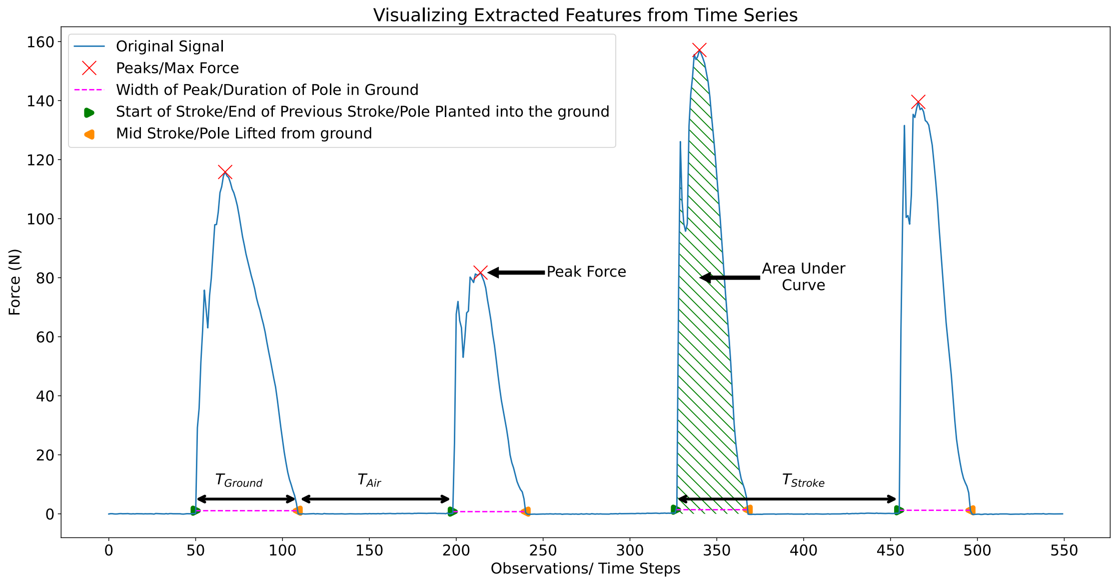


Figure 3.3: Zoomed-in view of force profile highlighting calculated features

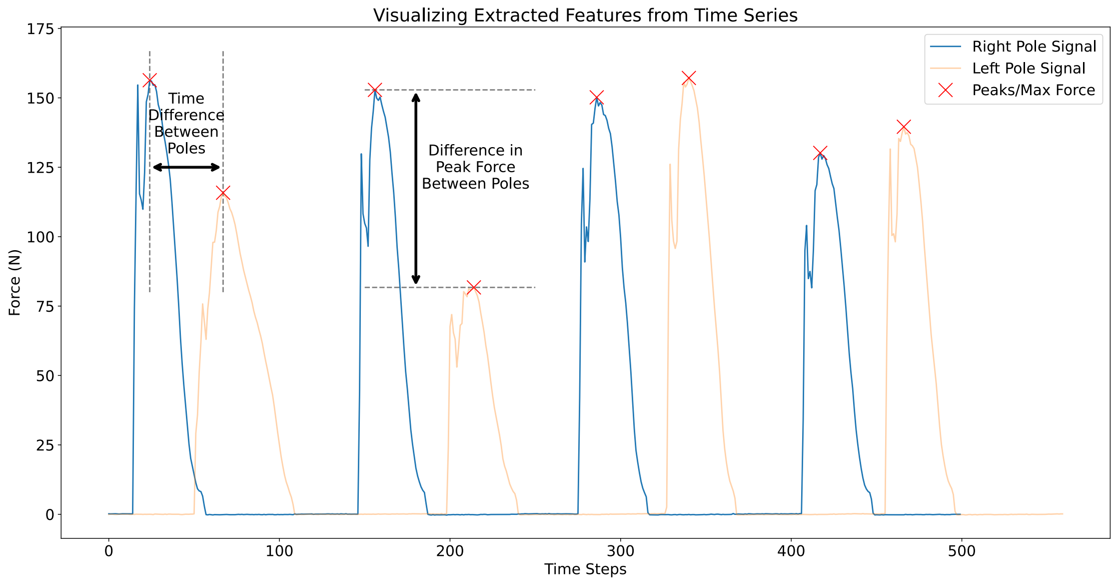


Figure 3.4: Superimposed force profile of both the pole sensors highlighting calculated features that use data from the left and right poles

The force data from the left pole of a skier on a treadmill is shown in Fig. 3.2 where each color represents the different techniques the skier performed over the time steps. Since the dataset is too large, this figure shows a small sample of data for visualization.

Upon zooming into each of the strokes closely, the force curves can be visualized individually as shown in Fig. 3.3. Here, the first four strokes of the left ski pole can be seen - characterized by four peaks, each representing the peak force the skier applied when planting the pole to the ground. Each cycle comprised of one stroke which was counted from the time the skis were planted to the ground to the next time the skis were planted to the ground again. This figure also highlights other properties of a stroke that were extracted as calculated features discussed in detail below. Features were extracted using the python signal processing library `SciPy`.

Ground Contact Duration (T_{ground})

Duration of time that the pole is in contact with the ground, such that a force is applied. The force is visible as a curve with a peak signifying the maximum force applied during the contact. The duration is highlighted in pink with green at ground contact start and orange at ground contact end in Fig. 3.3.

Air Time Duration (T_{air})

Duration of time that the pole is in the air, and no force is applied. The force is a straight flat line measured from the orange marker of previous ground contact end to the green marker of the next ground contacts start in Fig. 3.3.

Stroke Duration (T_{stroke}) = ($T_{ground} + T_{air}$)

Each stroke is one cycle of ground contact followed by air time. It is the duration of time between the beginning of two ground contacts.

Stroke Frequency ($1/T_{stroke}$)

It is the inverse of stroke duration.

Peak Force

Peak force is defined as the maximum force applied by the skier during each stroke when the ski pole is planted to the ground. This attribute is calculated individually for each pole and it is extracted by calculating the highest peak in a window of values supplied to a function in the `SciPy` library. The peak force can seen as a red 'x' in Fig. 3.3 on top of each of the curves.

Area Under Curve

Measured as the force area during the ground contact. It is calculated using the maximum peak force and ground contact duration. It is represented as the green hatched area in Fig. 3.3.

Time to other pole

It is defined as the time to the nearest peak for the opposite pole. It is measured from the peak force of the stroke on one pole to the peak force of the stroke on the other pole. It is visually represented in Fig. 3.4 where the force profiles of the left and right poles are superimposed on top of each other and the measured distance is depicted.

Difference in peak force between poles

It is defined as the difference in the peak forces between the peak of one pole to the nearest peak of the opposite pole. It is also measured from the peak force of the stroke on one pole to the peak force of the stroke on the other pole. Fig. 3.4 depicts the peak force difference between the two poles.

3.3 Exploratory Data Analysis

Once the data is cleaned and prepared, some exploratory data analysis is performed on the dataset before being input into the machine learning models. The idea was to understand if there are any patterns or if any observations can be made regarding the skiing techniques that could help understand the techniques better and by extension, probably the machine learning models as well. For the purposes of effective visualization, major outliers have been excluded from the dataset that is used to generate the plots. In order to understand if a machine learning model can generalize effectively, it could be useful to see if there are major differences in the skiing style between male and female skiers, or between skiing on a treadmill and skiing outdoors. In each of the figures in this section, the mean is represented by a dashed line in the middle of each distribution, and the dotted line above and below the mean represent the quartiles of the distribution.

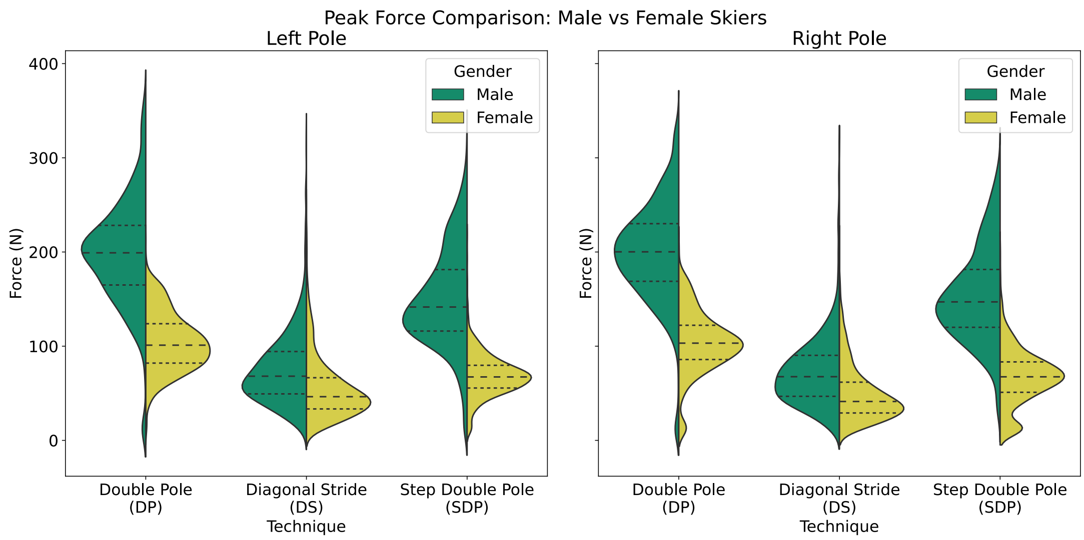


Figure 3.5: Comparison of Peak forces between male and female skiers across skiing techniques

Assessing Male Skiers versus Female Skiers

In this section, only data collected on the treadmill will be considered as there is no female skier data available from outdoors. Using the treadmill data also provides a relatively balanced split of data (6 male skiers and 5 female skiers).

Comparing Peak Forces Fig. 3.5 represents the distribution of peak force between male and female skiers for each technique. It is evident from the figure that the peak forces applied by the female skiers are generally much lower than those of the male skiers in Double Pole (DP) and Step Double Pole (SDP) techniques. This could likely be due to males in general having much more muscle mass on the upper body/arms, which would be most pronounced in double poling, as all force come from arms/upper body in that technique. There is even lesser difference in peak force applied by male and female skiers in Diagonal Stride (DS), represented by the smaller margin between their means.

Comparing Calculated Skiing Parameters From the discussion above (Fig. 3.5), a question that arises is - do male and female skiers have a similar way of skiing even though there is a difference in forces? In order to answer this, we can analyze the calculated skiing parameters to gain insight into their skiing styles, shown in Fig. 3.6.

Male and female skiers appear to have fairly similar duration of ground contact across all techniques (Fig. 3.6(a)). Although, looking at Fig. 3.6(b) for Double Pole (DP), female skiers have a lower air time duration and higher stroke frequency than male skiers. However, the overall duration of the stroke is quite similar for all skiers (Fig. 3.6(c)). Thus, irrespective of gender, the skiers appear to be skiing in a similar fashion. Additionally, as expected, the area under the curve (Fig. 3.6(e)) is considerably lower for females as they apply lower forces.

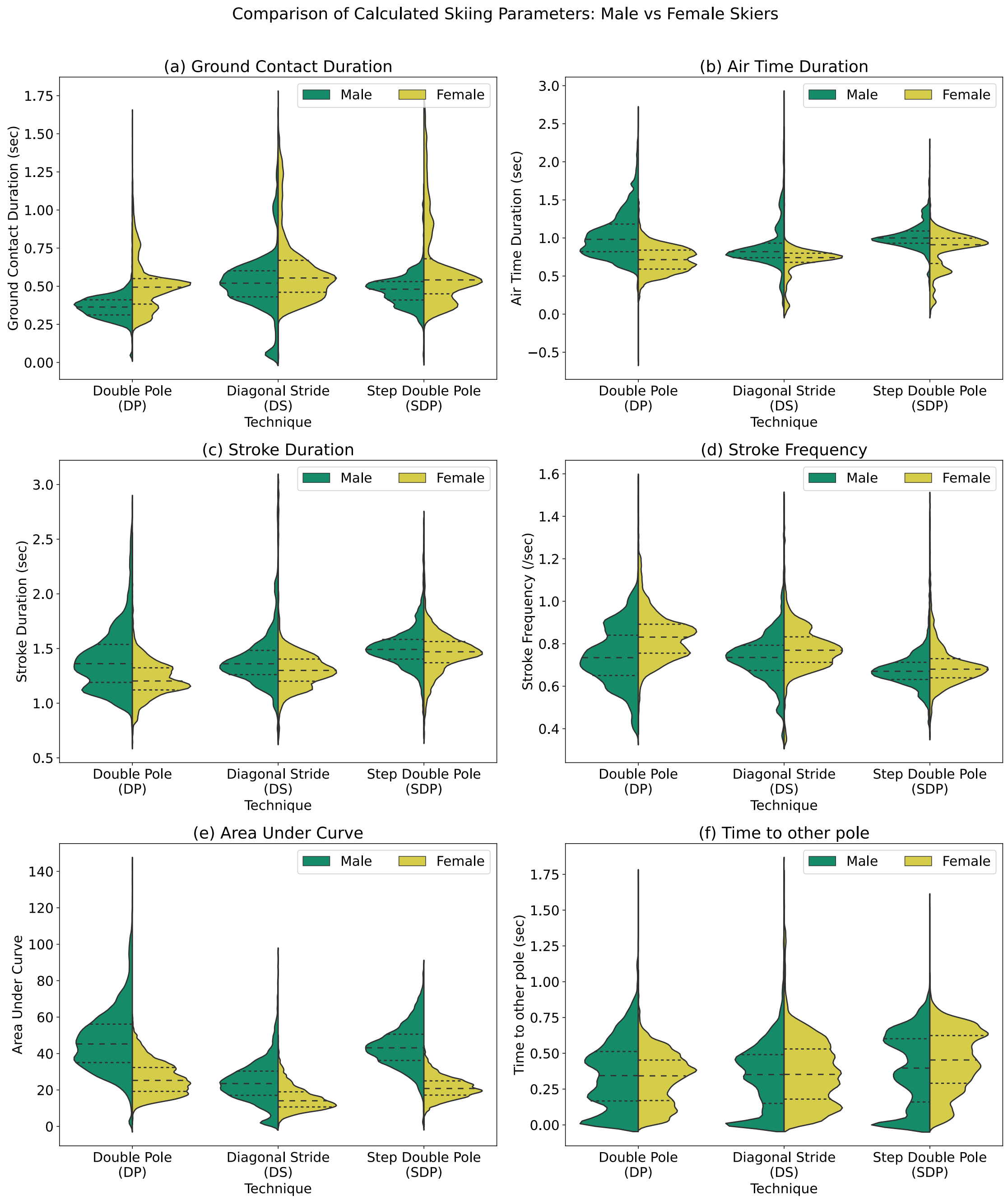


Figure 3.6: Comparison of calculated skiing parameters between male and female skiers across skiing techniques

Assessing Treadmill versus Outdoor Skiing

In this section, all collected data will be considered for analysis. It also provides a relatively balanced split of data (11 treadmill sessions and 10 outdoor sessions).

Comparing Peak Forces Fig. 3.7 represents the distribution of peak force while skiing on a treadmill and skiing outdoors. In this case, the peak forces applied while skiing outdoors is generally much higher than on a treadmill in Double Pole (DP) and Step Double Pole (SDP) techniques. External environmental factors could likely be the reason for applying higher forces as compared to skiing on a treadmill. As was the case earlier, here too, there is little difference in peak force applied in Diagonal Stride (DS) on a treadmill and outdoors.

Comparing Calculated Skiing Parameters Now, we analyze the calculated skiing parameters to gain insight into the treadmill and outdoor skiing styles, shown in Fig. 3.8. Fig. 3.8(a) shows that the ground contact duration is quite similar across the three techniques - represented by the small difference between their means. However, the air time duration is longer when skiing outdoors (Fig. 3.8(b)). These scenarios are likely due to the momentum the skier carries outdoors, in the open, as opposed to a limited track on a treadmill that requires continuous skiing in order to stay on the treadmill. Additionally, the treadmill data was always recorded on a range of slopes that were predominantly uphill. Skiing outdoors, in comparison has varied geography that would consist of uphill as well as gradual downhill and flat terrain. The skier would carry more momentum skiing on a flat or downhill terrain thus requiring the skier to perform the strokes less often, resulting in lesser ground contact time and a longer air time duration. This also translates into longer stroke duration skiing outdoors (Fig. 3.8(c)).

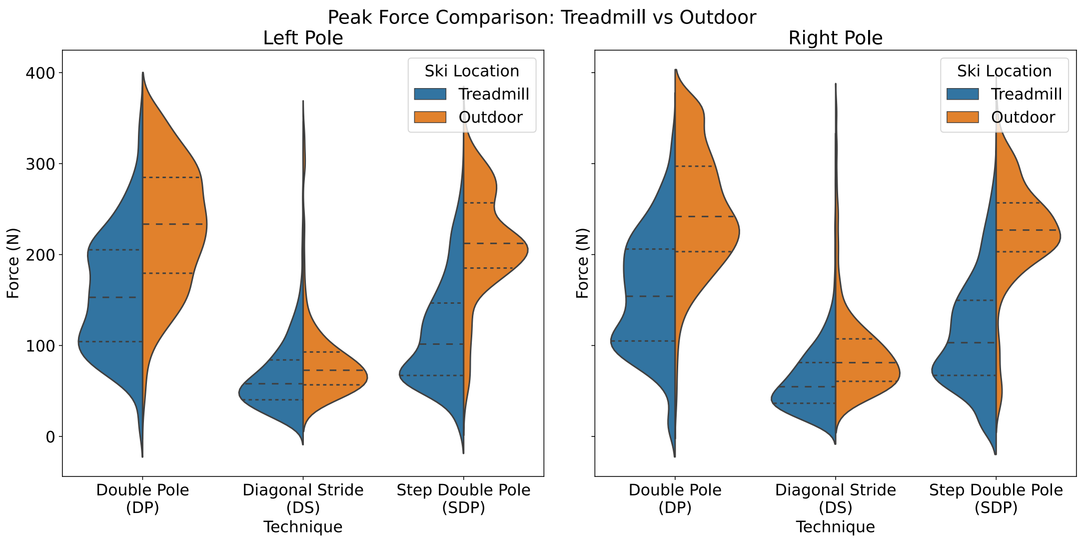


Figure 3.7: Comparison of Peak forces between skiing on a treadmill and skiing outdoors across skiing techniques

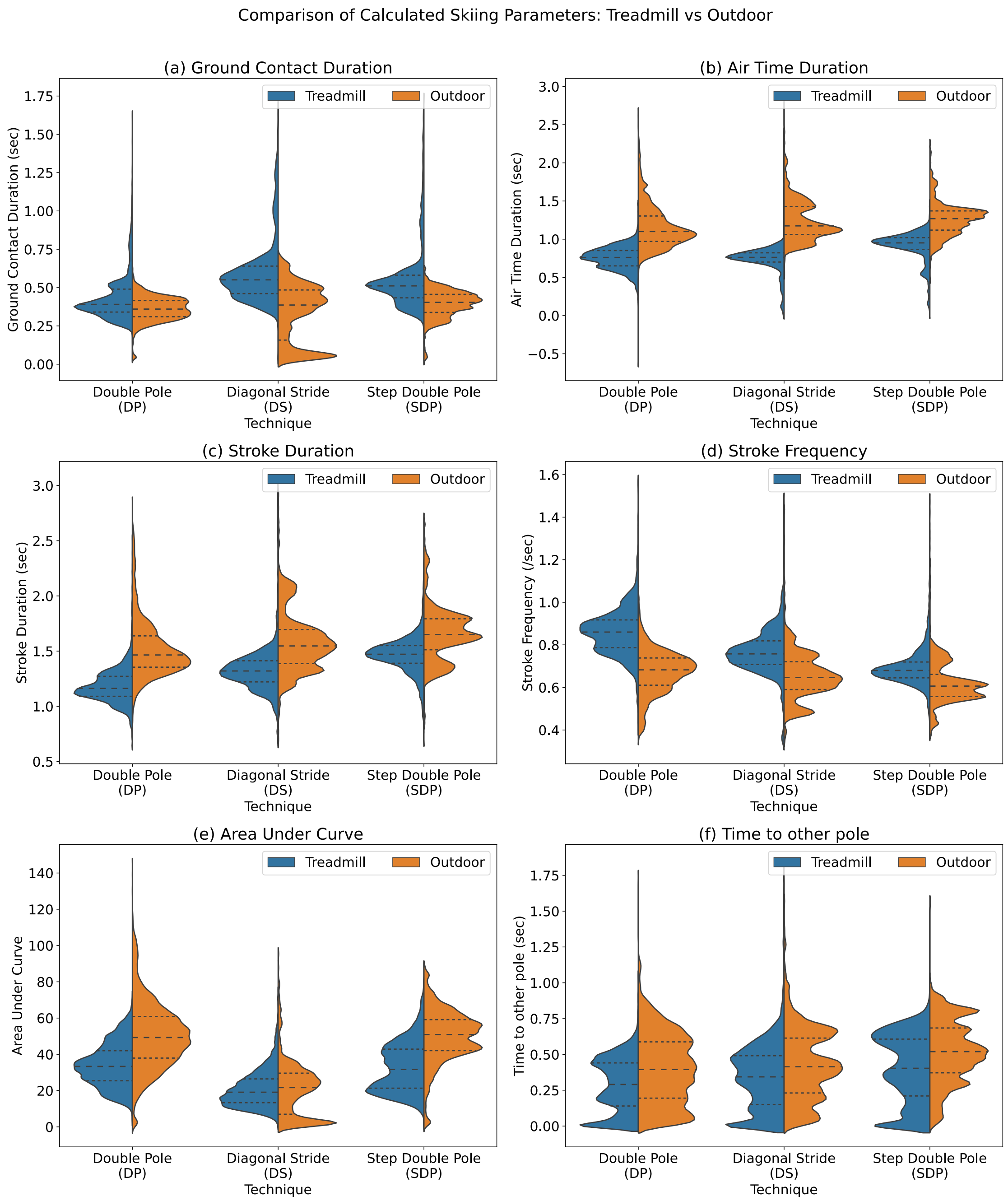


Figure 3.8: Comparison of calculated skiing parameters between skiing on a treadmill and skiing outdoors across skiing techniques

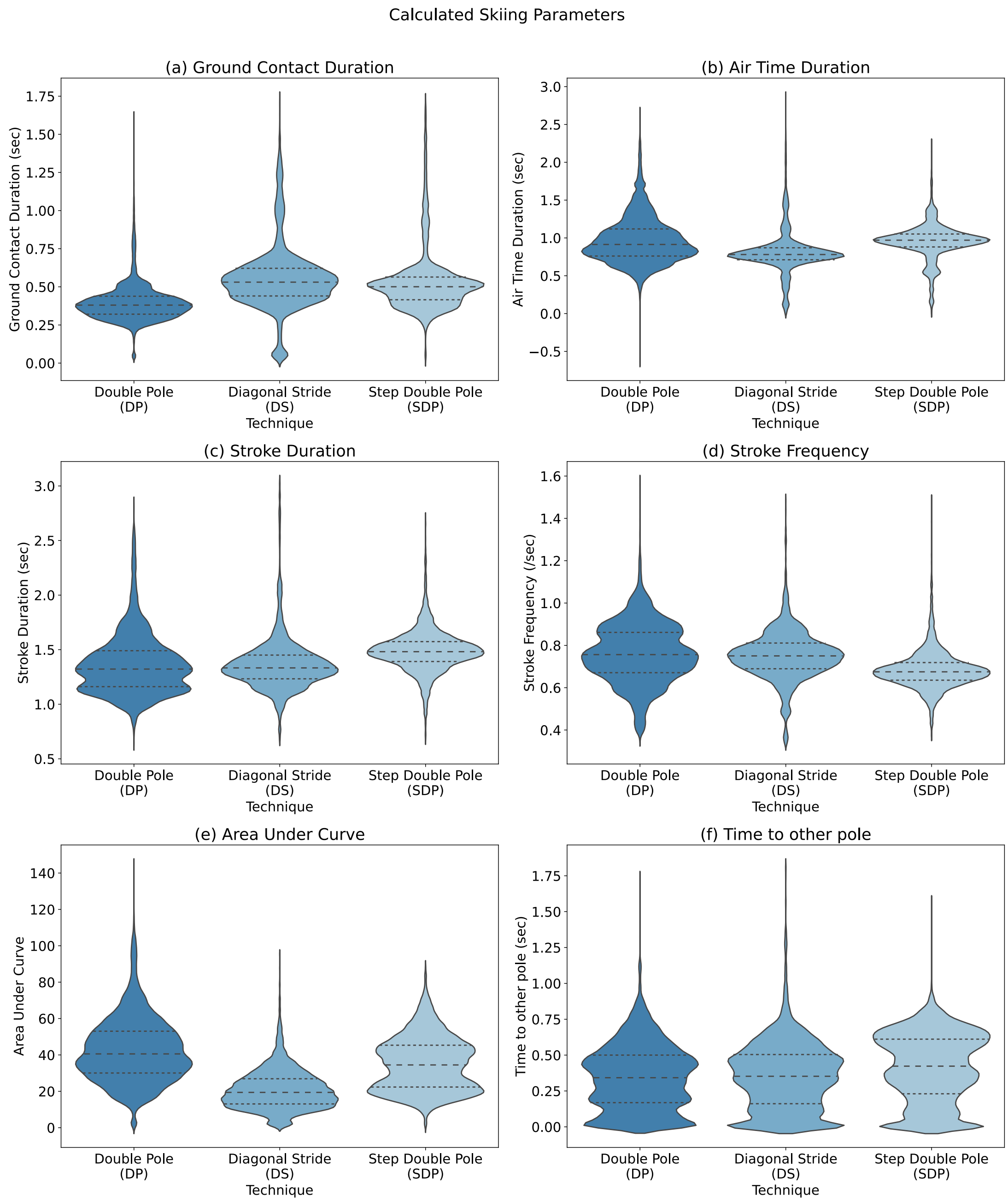


Figure 3.9: Distribution of calculated skiing parameters for all techniques

Assessing Difference in Skiing Techniques

Based on the analysis above, we can see that event though male and female skiers apply different amounts of force, their way of skiing is similar. On the other hand, there is a difference in the peak force and the style of skiing while skiing outdoors versus skiing on a treadmill. Thus, in order for a machine learning model to be able to generalize and predict skiing techniques performed anywhere, training the model on a mix of treadmill and outdoor data could be useful.

This section looks at the entire dataset together to observe any distinct patterns in the skiing parameters for the skiing techniques. Fig. 3.9(a) shows that the shortest ground contact time is during the Double Pole (DP) technique and is easily identifiable while the ground contact duration is relatively similar for the other two techniques. The air time duration in Fig. 3.9(b) is spread more for double pole than the other two techniques. Fig. 3.9(e) shows the significant difference in the area under the curve for each of the techniques which could be useful in predicting the technique.

3.4 Random Forest Classifier

This section outlines the random forest classification experiment for the identification of skiing techniques. The random forest model does not scale well for very large amounts of data, so using the force data from the poles directly would not be feasible. Instead, the model used the calculated skiing parameters that were defined in Section 3.2.2. The model is created using the `scikit-learn`[1] Python library.

3.4.1 Data Preparation

The input dataset has 8 feature columns that include the calculated parameters defined earlier - Ground Contact Duration, Air Time Duration, Stroke Duration, Stroke Frequency, Area Under Curve, Time to other pole, Peak Height Difference, Pole. Each row represents one stroke having values for each of the features and a corresponding label that needs to be predicted.

As discussed in Section 3.3, the model was trained on a mix of treadmill and outdoor sessions. The entire dataset was split into 80% Training data and 20% Test data. An equal number of sessions from treadmill data and outdoor data were selected to achieve a fair balance between the two. It is important to note that although the split is fairly even between treadmill and outdoor sessions, there was still an imbalance in the classes within each session - as the Double Pole (DP) technique was performed for a considerably longer duration than the other techniques. To account for this imbalance, the model made use of class weights. Table 3.3 summarizes the number of treadmill and outdoor skiing sessions used for training and testing.

Table 3.3: Number of sessions by skiing location used for evaluating Random Forest model

	Treadmill	Outdoor	**Total**
Training	9	8	**17**
Test	2	2	**4**

[1]`https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html`

3.4.2 Hyperparameter Tuning

Hyperparameters refer to the parameters that can be adjusted manually before the training process starts. They are used to optimize model performance according to the data that the model is being trained on. Hyperparameter tuning has significant benefits when it comes to random forests as they help regulate the size and complexity of the trees in the forest leading to improved memory management.

The following hyperparameters were selected for tuning:
1. Number of estimators : Range of number of trees in the random forest
2. Criterion : Function to measure the quality of a split
3. Maximum depth : Range of maximum number of levels/depth in a tree
4. Minimum samples to split : Range of Minimum number of samples required to split a node
5. Minimum samples in a leaf : Range of Minimum number of samples required at each leaf node
6. Bootstrap : Whether to use bootstrap samples or not when building trees

For this model, hyperparameter tuning was performed using the `RandomizedSearchCV` function from the `scikit-learn` library. This function selects a random combination of hyperparameters in an attempt to find the best performing set for the task. Although it may lead to high variance in the results during the search, it is also likely that it finds the optimal parameters because of the random search parameters. Since there were a lot of hyperparameters and their values for tuning, it was feasible to use a random search as it reduces the computation time as well. The above parameters were input to the function and 100 different combinations were sampled. 5-fold cross validation was performed on each combination and the parameters were evaluated based on the overall accuracy.

The hyperparameters that yielded the best results are summarized below:
1. Number of estimators : 500
2. Criterion : gini
3. Maximum depth : 10
4. Minimum samples to split : 4
5. Minimum samples in a leaf : 2
6. Bootstrap : True

The 'criterion' parameter specifies the function used to measure the quality of a split. 'Gini' refers to the Gini impurity which is the probability of incorrectly classifying a randomly chosen element in the dataset if it were randomly labeled according to the class distribution in the dataset[2]. The gini gain is calculated by subtracting the weighted impurities of the branches from the original impurity. It is possible to choose the best split in a tree by maximizing the gini gain. The model was saved as a `binary` file with `.sav` format using the `pickle` library that saved

[2] `https://victorzhou.com/blog/gini-impurity/`

the entire model architecture. This made it easy to reload the model and re-use it when required.

3.5 LSTM Neural Network Classification

This section defines the experimental setup for the identification of skiing techniques using LSTM neural networks. For this model, the force values data from both the ski poles was used as input to predict the skiing techniques. The neural network model is created using the `TensorFlow` Python library.

3.5.1 Data Preparation

The input dataset is 2-dimensional, but LSTM requires a 3-dimensional input in the form of `(Samples, Time Steps, Features)`. Here, a sample is a sequence or batch of one or more time steps. Each time step is one row in our input dataset and there are two feature columns representing force, one for each hand (left and right). In order for the LSTM to read input data, the 2D data is converted to 3D data by creating batches of the input data. By manual inspection of the input data, it was determined that the batch size should be 150 consecutive time steps, each batch representing one stroke as best as possible. The label for each of the sequences was determined by a majority vote that assigned the label of the most frequently occurring label in the sequence to it. The result was a 3D input dataset of shape `(n, 150, 2)` with n sequences, of 150 time steps each having 2 features.

The same split and the same dataset was carried forward from the random forest classifier to train and evaluate the neural network model as well, so that the results can be compared effectively. However, for the neural network model, in addition to a training and test dataset, there was also a validation dataset created that was ~10% of the training dataset to evaluate performance of the model during training. Similar to the random forest model, `class weights` are used to balance the labels in the dataset. Table 3.4 summarizes the number of treadmill and outdoor skiing sessions used for training, validation and testing.

Table 3.4: Number of sessions by skiing location used for evaluating LSTM neural network

	Treadmill	Outdoor	**Total**
Training	8	7	**15**
Validation	1	1	**2**
Test	2	2	**4**

3.5.2 Hyperparameter Tuning

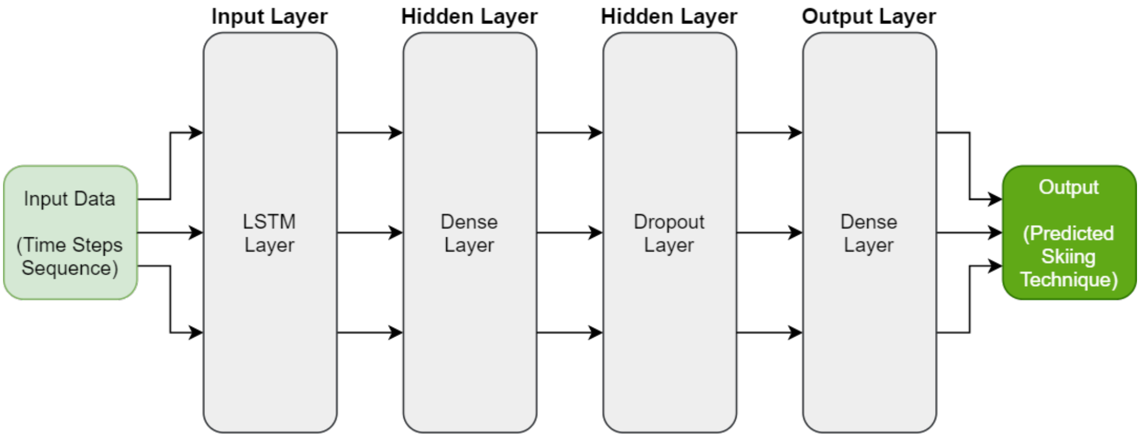


Figure 3.10: Distribution of calculated skiing parameters for all techniques

Fig. 3.10 represents the LSTM model used for prediction of skiing techniques. It consists of an LSTM input layer connected to a fully connected layer and a dropout layer followed by a fully connected output layer. In order to optimize the model and achieve maximum performance, hyperparameter tuning was performed and visualized using `Tensorflow's`[3] visualization toolkit - `TensorBoard`.

The following hyperparameters were selected for tuning:

1. LSTM Layer : Number of neurons
2. Fully Connected Hidden Layer : Number of neurons
3. Dropout Layer : Dropout rate
4. Batch Size : Number of samples to evaluate before updating the model
5. Epochs : Number of times the model is executed to learn about the data
6. Optimizer : Method used to update the model weights and reduce losses

The range of values for each of the hyperparameters was selected based on trial and error and subsequent narrowing down of the values was done based on the model performance in terms of accuracy. Multiple combinations of hyperparameters were selected and run in phases to save on execution time instead of running hundreds of unique combinations every run. Fig. 3.11 represents one such example where a set of hyperparameters from above were tuned. For all combinations, the model uses a `Softmax` activation function in the output layer and a `categorical cross entropy` loss function since it is a multi class classification problem.

[3] https://www.tensorflow.org/

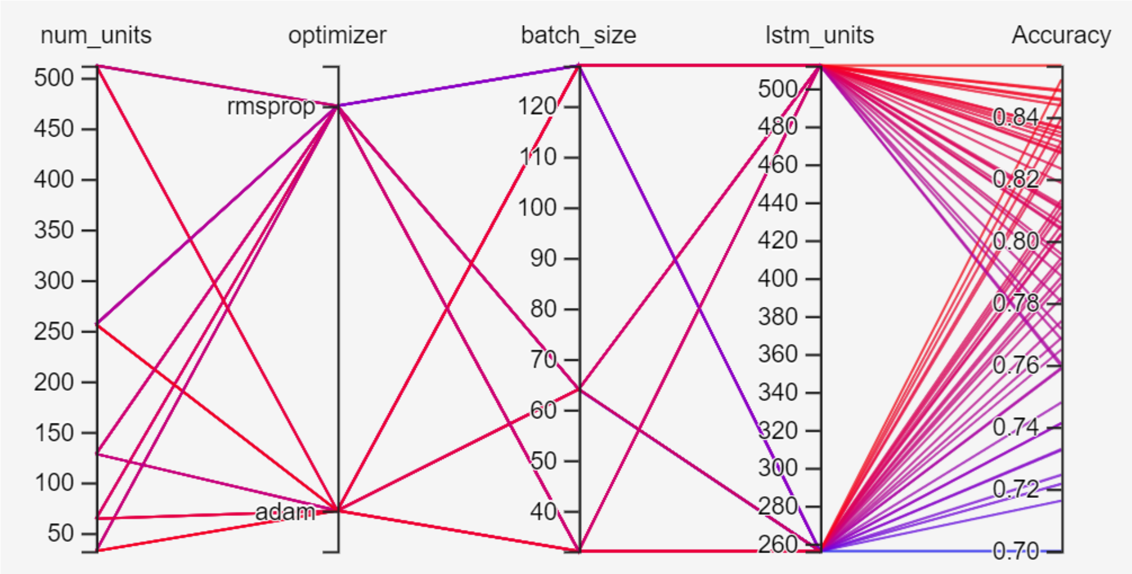


Figure 3.11: `TensorBoard` Screenshot showing the accuracy of a model using different value combinations of hyperparameters. `num_units` represents the number of neurons in fully connected hidden layer, `lstm_units` represents the number of neurons in LSTM layer.

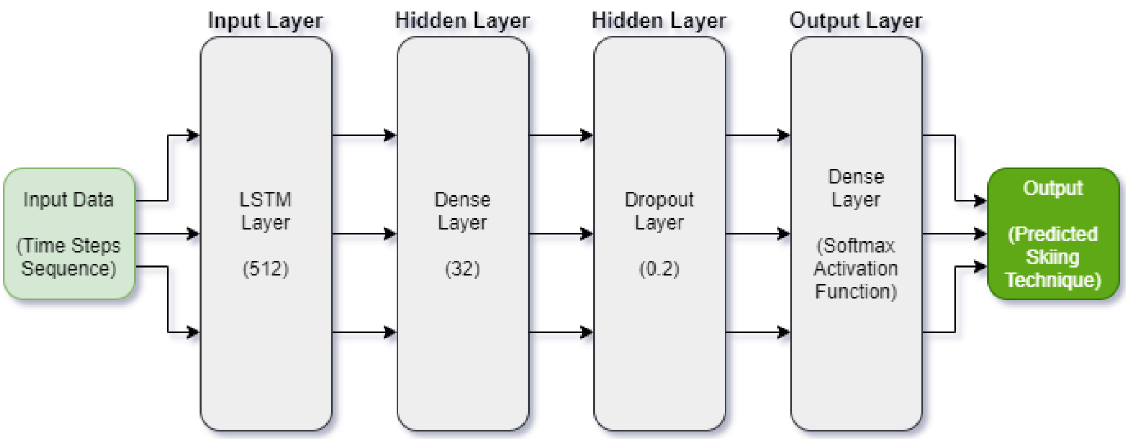


Figure 3.12: Architecture of the LSTM Neural network showing the input and output in green and the neural network layers in gray. The number of neurons and dropout rate are mentioned in brackets.

After multiple iterations of hyperparameter tuning, the model represented in Fig. 3.12 performed the best and was finalized for identification of skiing techniques. The hyperparameters are summarized below:

1. LSTM Layer : 512 Neurons
2. Fully Connected Hidden Layer : 32 Neurons
3. Dropout Layer : 0.2 Dropout rate with 32 Neurons
4. Batch Size : 64
5. Epochs : 50

6. Optimizer : Adam Optimizer
7. Loss function : Categorical cross entropy

The model was saved as a `HDF5` file with `.h5` format that saved the entire model architecture, weights values and `compile()` information. This enables the model to be reused for future research purposes and potentially for transfer learning by building on this treating it as a pre-trained model.

4

Results

This chapter describes the setup to carry out the experiments and share the results. It details the steps taken to evaluate the machine learning models and shares the prediction results of each model with their respective evaluation metrics.

4.1 Experiment Setup

Initial creation of '.csv' files was performed in MATLAB[1] on a Windows laptop with an Intel Core i3-5005u 2.0 GHz processor and 8GB of memory. All the experiments were performed on Google Colaboratory Pro[2] and its GPU resources were utilized while evaluating the neural network model.

The following steps were taken for both the models while evaluating the model performance and recording the results. Any deviation from the below or additional steps taken specific to a machine learning model has been described in their respective sections.

- The models were trained and tested on a mix of treadmill and outdoor datasets using 10-fold cross-validation.
- The results were recorded for the entire test data set, and separately for the treadmill component and outdoor component of the test dataset. This was done to understand whether the overall test prediction was indicative of predicting skiing techniques in both the scenarios or was heavily skewed due to its predictive performance in either the treadmill or outdoor scenario.
- An additional step to refine the predicted results was also performed. From preliminary analysis, it was observed that the predicted labels for the strokes/-time step sequences next to each other were switching between values frequently and such high variation in technique in a small window of observations is not realistically possible, as a skier will not be switching between gears so often. So, the predicted labels were further refined by applying a majority filter.

  Single Majority Filter: In the single majority filter, a window size of 5 observations was defined, where for every label, the filter calculates the most commonly occurring label around the current label and updates the current label to that value. This way, if there are outliers within a range of observations, they could be 'corrected' using the majority filter.

[1] `https://se.mathworks.com/products/matlab.html`

[2] `https://colab.research.google.com/notebooks/intro.ipynb?utm_source=scs-index`

Multiple Majority Filter: A second variation of the majority filter was also used, which made multiple passes on the predicted labels using windows of varying lengths. This filter resulted in a cleaner grouping of the predicted labels and eliminated any remaining outliers from the Single Filter.

- For each of these experiments, a confusion matrix with the accuracy score of each label was calculated. This helped in understanding the patterns in prediction and identify potential areas of improvement. Additionally, the profile of the true skiing technique labels was plotted against the predicted values to visualize the quality of predictions made.

4.2 Baseline Model

We begin by considering a simple baseline model to help assess whether our more complex solution with hyperparameter tuning performs better than the baseline model or not. The baseline model we consider uses information from Fig. 3.9 to visually select features that can help distinguish between the skiing techniques. The Double Pole technique is the most dominant class in the dataset followed by Step Double Pole and then Diagonal Stride in terms of number of strokes. Thus, for this classification problem we will begin by labeling all strokes as Double Pole by default and then work our way towards predicting the step double pole and diagonal stride strokes from that dataset. Thus anything that is not predicted as SDP or DS is automatically DP by default.

Upon visual inspection we can see from Fig. 3.9b,d that the SDP technique has a longer air time duration and lower stroke frequency than DS. Similarly, we can see from Fig. Fig. 3.9a,b and e that DS has a longer ground contact duration, shorter air time duration and considerably lower area under curve than DP and SDP. Using these calculated features, it should be possible to predict the skiing techniques by filtering the dataset based on the values. Values between the 25^{th} and 75^{th} quantile ± 0.5 IQR were considered for each feature and the rows that satisfied these criteria for SDP and DS were labeled respectively in that order.

Table 4.1 summarizes the performance of the baseline model. The baseline model has an accuracy of 61.2%, a balanced accuracy of 58.8% and an MCC score of 0.370. The confusion matrix in Fig. 4.1 depicts the prediction accuracy using the baseline model for each skiing technique where a high number of strokes were incorrectly classified as Double Pole, mostly because it was the default value and due to the limited range of values considered to identify the other two techniques. However, it is interesting to note that even in this baseline model we were able to distinguish between DS and SDP quite effectively with DS being predicted with 58% accuracy. Additionally, the DP and SDP technique were incorrectly classified as each other most frequently, which is in line with the expectation as the two techniques have some common movements. Neither of these techniques were confused significantly in with Diagonal Stride, which is promising and can potentially be improved further.

Table 4.1: Evaluation Metrics for the Baseline Random Forest Model

Accuracy	Balanced Accuracy	MCC Score
0.612	0.588	0.370

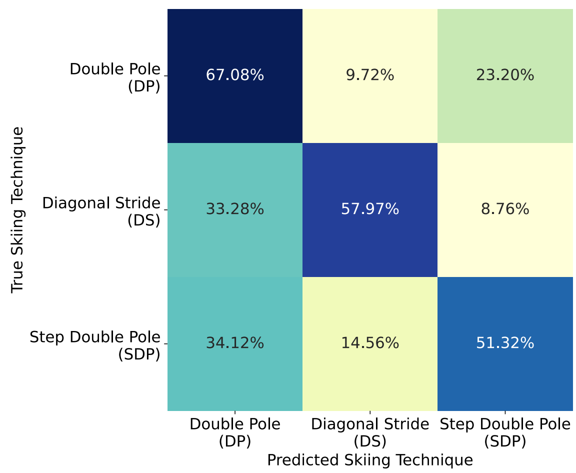


Figure 4.1: Confusion matrix for prediction accuracy of the baseline random forest model

4.3 Random Forest Classifier

Feature Importance

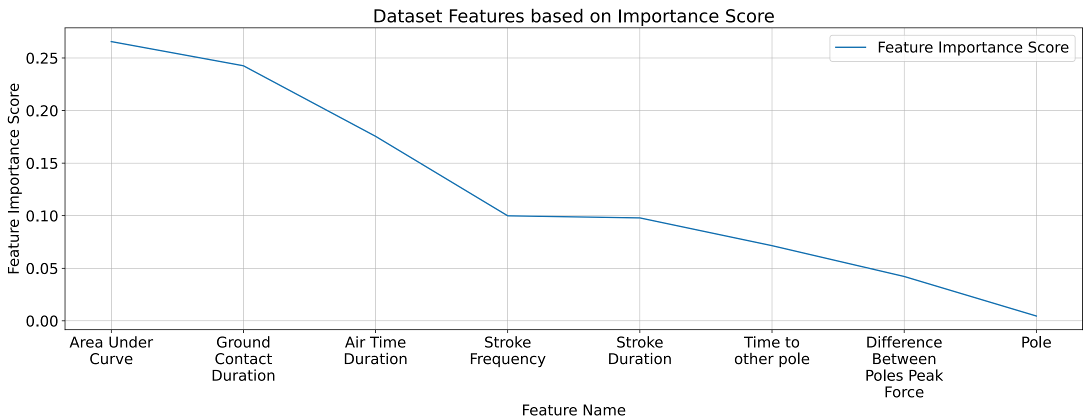


Figure 4.2: Feature importance score for the entire feature set used to train the random forest model

10-fold cross-validation was run on the random forest model and the results were recorded. Since random forest classifiers also calculate feature importance as a measure, it was interesting to visualize it as in Fig. 4.2 and evaluate if there are any features that could potentially be reduced. The area under the curve, the ground contact and air time duration were determined to be the most important features of the model while the difference in the peak forces between the poles and the pole label were the least important features. It is likely that the lesser important features could be adding noise to the data resulting in reduced performance of the model. For instance, the pole label would not be relevant as the same movement is done by both arms in synchronization and thus generating similar force profiles for either pole. Similarly, the difference in the peak forces between the poles would not be of importance as similar force is being exerted by the skier on each pole, making the difference in the peak forces insignificant. Thus, in addition to evaluating the random forest classifier, an additional step of reducing the feature set was done and performance evaluated. The results for both these experiments were recorded to observe any difference/improvement in performance.

4.3.1 Model with complete feature set

As mentioned above, the first experiment was to run the random forest model using the entire feature set and record the results. The results from the 10-fold cross validation were recorded for the entire test data set and for the treadmill and outdoor components of the test dataset as well. The results are summarized in the below tables. The values denoted for every evaluation metric are the average of the 10-fold cross validation with the standard deviation within the brackets.

Cross-validation summary

Table 4.2 summarizes the results from the entire test dataset. The model yielded 72.7% overall accuracy and 70.7% balanced accuracy. The standard deviation was however, large and the values seemed to vary highly across different cuts of the data during cross validation. The MCC score for each of the scenarios was not very high. On passing the predicted labels through a single majority filter, there was an average improvement of ~3-4% across all evaluation metrics, with similar standard deviation as without the filter. Passing the predicted labels through multiple majority filters yielded similar performance with minimal gains compared to the single filter. However, the advantage of a multi filter is that it achieves a cleaner grouping visually by grouping even the minor outliers that were left by the single filter into larger uniform groups. This is explained further in the subsequent sections visually.

Table 4.2: Cross-validation summary for Random Forest Model. Table depicts average across folds, with standard deviation in brackets.

	Model Results Only	Model Results Passed Through Single Filter	Model Results Passed Through Multiple Filters
Accuracy	0.727 (0.056)	0.760 (0.058)	0.769 (0.053)
Balanced Accuracy	0.707 (0.049)	0.740 (0.050)	0.740 (0.047)

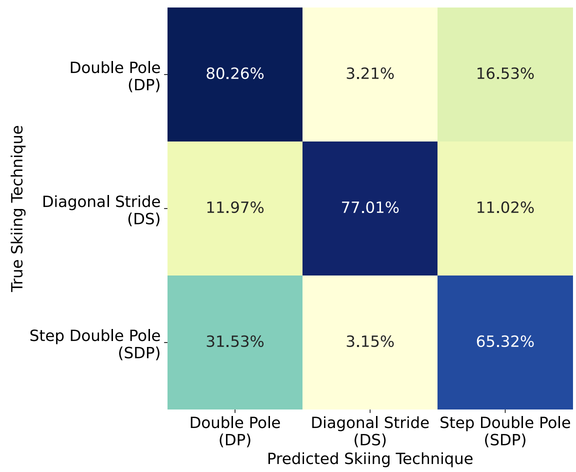
MCC Score	0.573 (0.078)	0.626 (0.081)	0.638 (0.072)

Visualizing model performance

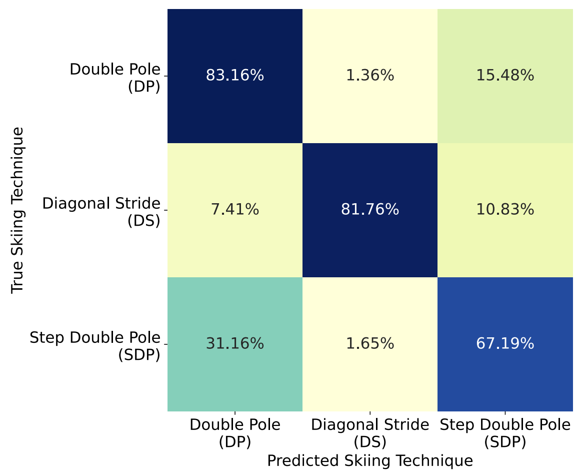
The results were visualized as below to look for any additional patterns and observations that can be made. Confusion matrices (Fig. 4.3) were used to assess the classification performance between skiing techniques. Comparison plots (Fig. 4.4) were used to visualize the profile/order the skiing techniques performed and superimpose the predicted labels to see how well the classification performed. Although cross-validation had certain folds with very high accuracy than the average, but for generalization purposes, the folds that performed nearest to the average value are visualized below.

The confusion matrices are given in Fig. 4.3 for the entire test data set representing classification accuracy for each experiment. The model classified Double Pole and Step Double Pole skiing techniques the most accurately. In these plots, Diagonal stride was confused with Step Double Pole the most frequently. This may not necessarily be the case in general, but would require further investigation of confusion matrices from every fold evaluated. This was followed by fairly equal incorrect classifications between DP and SDP.

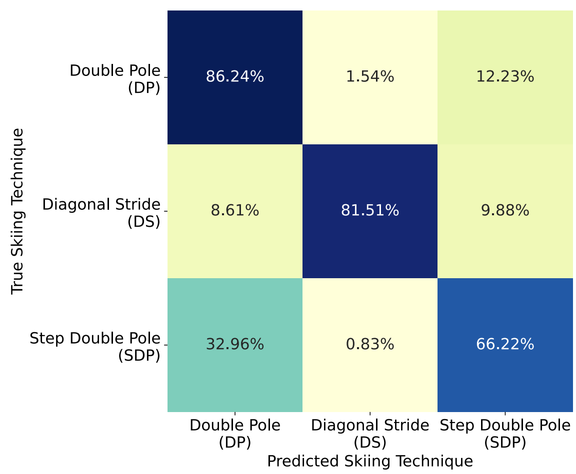
The comparison of true skiing techniques and corresponding predictions are given in Fig. 4.4. The predictions from the model are depicted in Fig. 4.4a where a lot of predicted labels are spread across different techniques in a small window. As this is not possible in practice, it was important to try to clean some of these labels to group them better. Passing the predicted labels through the single filter yields a cleaner grouping of skiing techniques as visible in Fig. 4.4b. Using the majority filter on the predicted labels removes even the other outliers from Fig. 4.4b and improves them as in Fig. 4.4c. It was also interesting to see the strokes between 4000-5000 in this figure where SDP was grouped well, but seemed to be offset by a few stroked from the true labels.



(a) Model Results Only

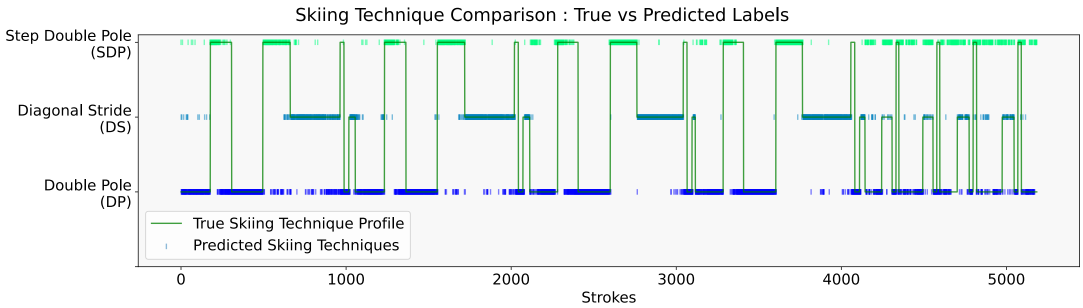


(b) Model Results Passed Through Single Filter

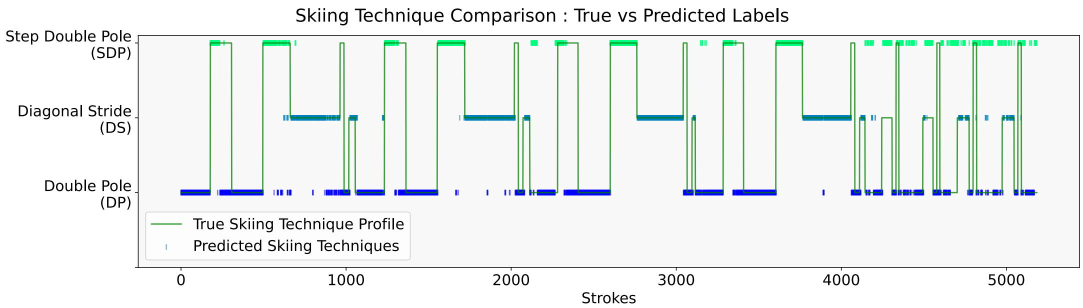


(c) Model Results Passed Through Multiple Filters

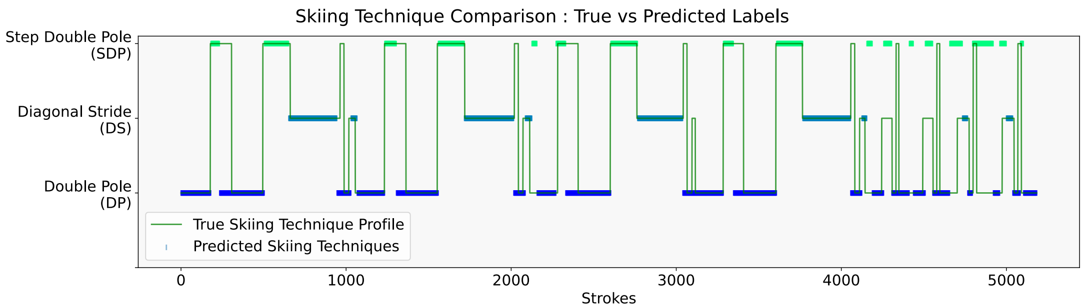
Figure 4.3: Confusion matrix for accuracy of Random Forest model



(a) Model Results Only



(b) Model Results Passed Through Single Filter



(c) Model Results Passed Through Multiple Filters

Figure 4.4: Comparison of true skiing techniques vs. predicted skiing techniques

Assessing Model Performance for Treadmill and Outdoor Data

In this section, we will use the same model to test separate datasets comprising of only treadmill data and only outdoor data, using 10-fold cross validation.

Table 4.3 summarizes the results from just the data that was collected on the treadmill. The random forest performed really well on this part of the dataset, yielding 89.1% overall accuracy and 88.7% balanced accuracy, implying that the classification across all techniques was more accurate. Even the standard deviation was comparatively small, varying ~4% for all evaluation metrics across the experiments. The MCC score for each of the scenarios was closer to 1. As the model had classified accurately, passing the predicted labels through a single or multiple majority filter, resulted in a further improvement of ~2-3% across all evaluation metrics.

Table 4.4 summarizes the results from just the data that was collected outdoors. The random forest yielded 84.6% overall accuracy and 74.2% balanced accuracy, with fairly high standard deviation of ~7-10% in accuracy and balanced accuracy but a low deviation in MCC score (~6%). The significantly lower balanced accuracy when compared to the overall accuracy implies that the model was unable to predict all the labels equally well. Passing the predicted labels through a single majority filter improved the evaluation metrics by ~2%. The multiple majority filter, however, resulted in a decrease in the evaluation metrics. This was because the larger window sizes of the multiple majority filter caused a lot of the labels from the minority classes to be re-labeled to the dominant class, i.e., Double Pole. This resulted in the labels grouped continuously without outliers, but at the expense of accuracy. Comparing tables 4.3 and 4.4, we can see that outdoor data has more variability compared to treadmill data potentially since it was not collected in a controlled environment like the treadmill data.

Table 4.3: Cross-validation summary for Random Forest Model on Treadmill Test Data. Table depicts average across folds, with standard deviation in brackets.

	Model Results Only	Model Results Passed Through Single Filter	Model Results Passed Through Multiple Filters
Accuracy	0.891 (0.041)	0.920 (0.040)	0.912 (0.031)
Balanced Accuracy	0.887 (0.045)	0.917 (0.043)	0.906 (0.033)
MCC Score	0.837 (0.060)	0.883 (0.057)	0.871 (0.044)

Table 4.4: Cross-validation summary for Random Forest Model on Outdoor Test Data. Table depicts average across folds, with standard deviation in brackets.

	Model Results Only	Model Results Passed Through Single Filter	Model Results Passed Through Multiple Filters
Accuracy	0.846 (0.079)	0.867 (0.077)	0.808 (0.064)
Balanced Accuracy	0.742 (0.098)	0.768 (0.102)	0.557 (0.063)
MCC Score	0.678 (0.142)	0.721 (0.141)	0.552 (0.118)

4.3.2 Model with reduced feature set

As mentioned in the beginning, using the feature importance plot (Fig. 4.2), it was decided to remove the two lowest important features from the dataset and evaluate the model performance to gauge any improvements. Thus, the first six features from the plot were considered for the below experiment.

Cross-validation summary

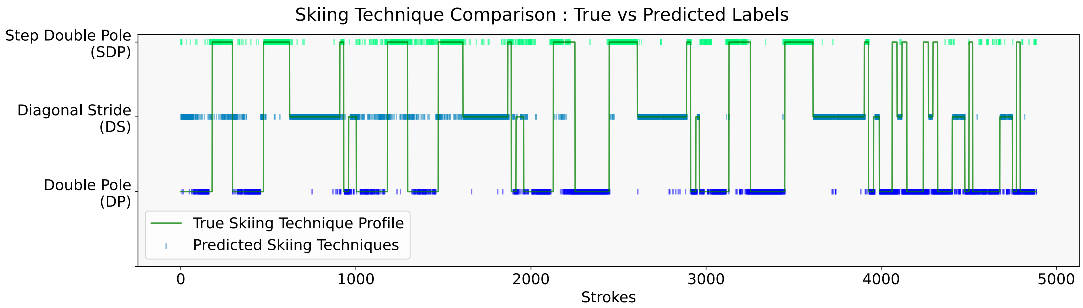
Fig. 4.5 summarizes the results from the entire test dataset. There is an improvement in overall accuracy by 1.7% to 74.4% and increase in balanced accuracy by 2.4% to 73.1%. The standard deviation was lower by 1% as well varying between ~3.5-4.5%. The MCC score, however, still remained low but was an improvement from the model using the entire feature set by 0.025. Passing the predicted labels through a single and multiple majority filter, resulted in an average improvement of ~3-4% from the results of the model across all evaluation metrics. Passing the predicted labels through multiple majority filters yielded similar performance with minimal gains compared to the single filter.

Table 4.5: Cross-validation summary for Random Forest Model with reduced features. Table depicts average across folds, with standard deviation in brackets.

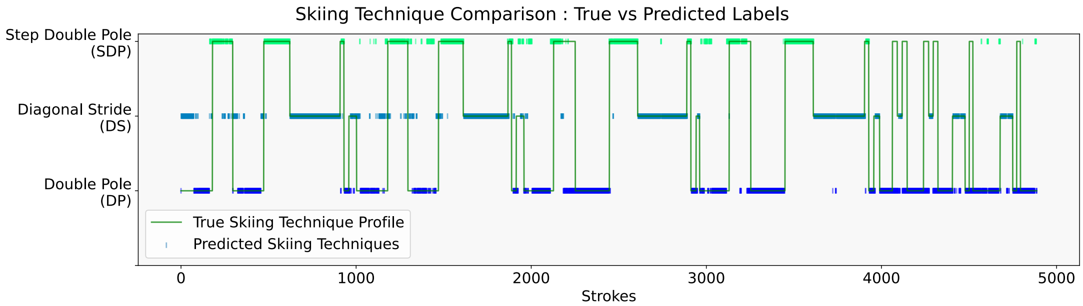
	Model Results Only	Model Results Passed Through Single Filter	Model Results Passed Through Multiple Filters
Accuracy	0.744 (0.045)	0.778 (0.042)	0.784 (0.028)
Balanced Accuracy	0.731 (0.036)	0.764 (0.035)	0.757 (0.025)
MCC Score	0.598 (0.076)	0.648 (0.075)	0.651 (0.056)

Visualizing model performance

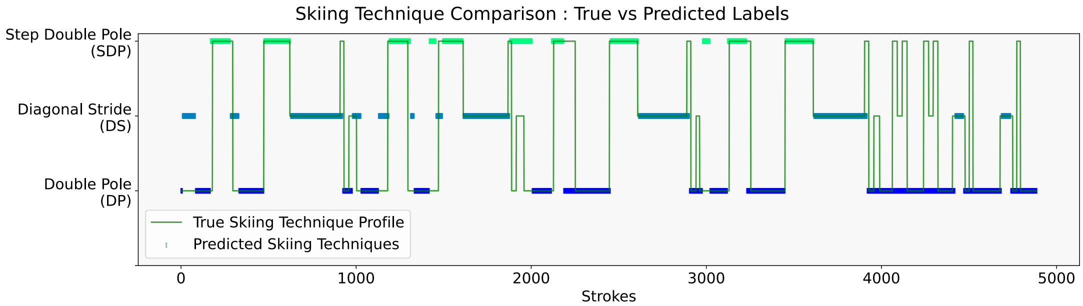
Confusion matrices (Fig. 4.6) were used to assess the classification performance between skiing techniques. Comparison plots (Fig. 4.5) were used to visualize the profile/order the skiing techniques performed and superimpose the predicted labels to see how well the classification performed. Similar to the previous experiment, the folds that performed nearest to the average value are visualized below. In Fig. 4.6 the classification accuracy of Double Pole and Diagonal Stride skiing techniques was predicted most accurately. The model was confused the most often between DP and SDP, with SDP being predicted least accurately. Using knowledge of the skiing techniques, there was some truth to the fact that the model was getting confused between DP and SDP as they have similar movements except for a kick in SDP. Looking at the high accuracy scores of DP and DS, but low score for SDP, one can understand why this model had such low balanced accuracy.



(a) Model Results Only

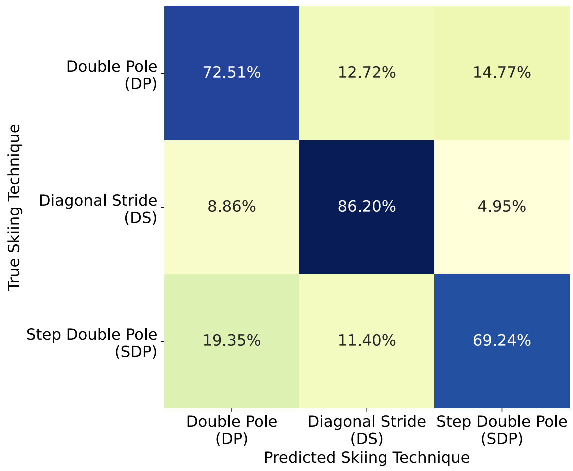


(b) Model Results Passed Through Single Filter

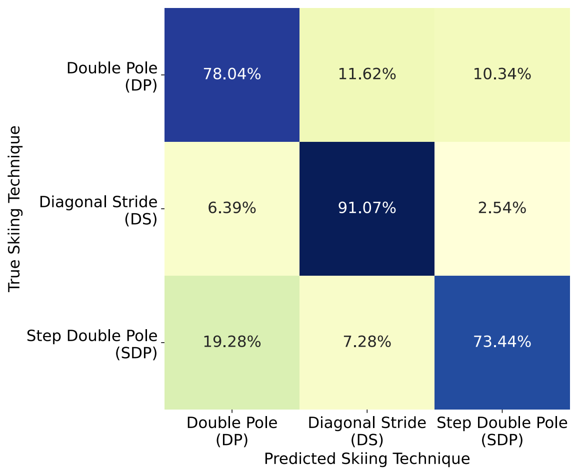


(c) Model Results Passed Through Multiple Filters

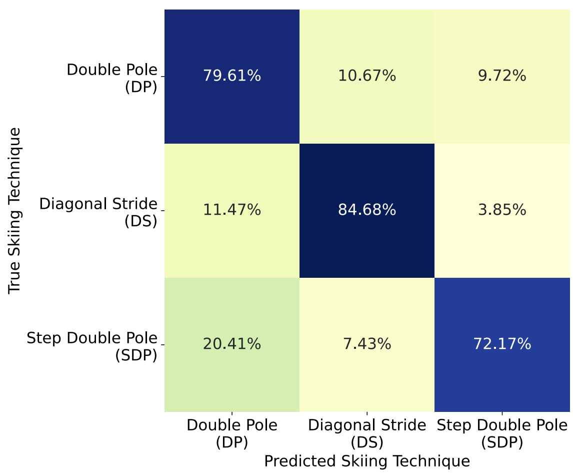
Figure 4.5: Comparison of true skiing techniques vs. predicted skiing techniques



(a) Model Results Only



(b) Model Results Passed Through Single Filter



(c) Model Results Passed Through Multiple Filters

Figure 4.6: Confusion matrix for accuracy of Random Forest model with reduced feature set

Assessing Model Performance for Treadmill and Outdoor Data

In this section, we will use the same model to test separate datasets comprising of only treadmill data and only outdoor data, using 10-fold cross validation.

Table 4.6 summarizes the results from just the data that was collected on the treadmill. The random forest performed really well on this part of the dataset, yielding 89.8% overall accuracy and 89.7% balanced accuracy, improving upon the accuracy from the previous random forest by 1%. However, the standard deviation was significantly lower than the earlier model (↓ by 2%). The MCC score was similar and on passing the predicted labels through a single or multiple majority filter, resulted in a further improvement of ~2-4% across all evaluation metrics.

Table 4.7 summarizes the results from just the data that was collected outdoors. The random forest yielded 85.8% overall accuracy (↑ by 1%) and 70.5% balanced accuracy (↓ by 4%), with slightly lower standard deviation of ~4-7% in accuracy and balanced accuracy but a high deviation in MCC score (~10%). As earlier, the significantly lower balanced accuracy when compared to the overall accuracy implies that the model was unable to predict all the labels equally well. Passing the predicted labels through a single majority filter improved the evaluation metrics by ~2%. The multiple majority filter, however, resulted in a decrease in the evaluation metrics.

Table 4.6: Cross-validation summary for Random Forest Model with reduced features on Treadmill Test Data. Table depicts average across folds, with standard deviation in brackets.

	Model Results Only	Model Results Passed Through Single Filter	Model Results Passed Through Multiple Filters
Accuracy	0.898 (0.023)	0.931 (0.021)	0.916 (0.020)
Balanced Accuracy	0.897 (0.025)	0.929 (0.022)	0.911 (0.021)
MCC Score	0.847 (0.035)	0.896 (0.031)	0.875 (0.029)

Table 4.7: Cross-validation summary for Random Forest Model with reduced features on Outdoor Test Data. Table depicts average across folds, with standard deviation in brackets.

	Model Results Only	Model Results Passed Through Single Filter	Model Results Passed Through Multiple Filters
Accuracy	0.858 (0.046)	0.880 (0.044)	0.835 (0.019)
Balanced Accuracy	0.705 (0.079)	0.718 (0.093)	0.565 (0.036)
MCC Score	0.678 (0.103)	0.722 (0.103)	0.606 (0.055)

4.4 LSTM Neural Network

Using the data setup mentioned in 3.5.1, 10-fold cross-validation was run on the tuned LSTM model and the results were recorded. A similar setup to that of the random forest is used in this case and results were recorded for a test dataset and its treadmill and outdoor components separately as well.

Cross-validation summary

Table 4.8 summarizes the results from the entire test dataset. The model yielded 86.2% overall accuracy and 85.6% balanced accuracy implying good accuracy scores across all skiing techniques. The standard deviation was low as well (~2-3%). The MCC score was relatively high and further improved in the filter results. On passing the predicted labels through a single and multiple majority filter, there was an improvement of ~4-5% in accuracy.

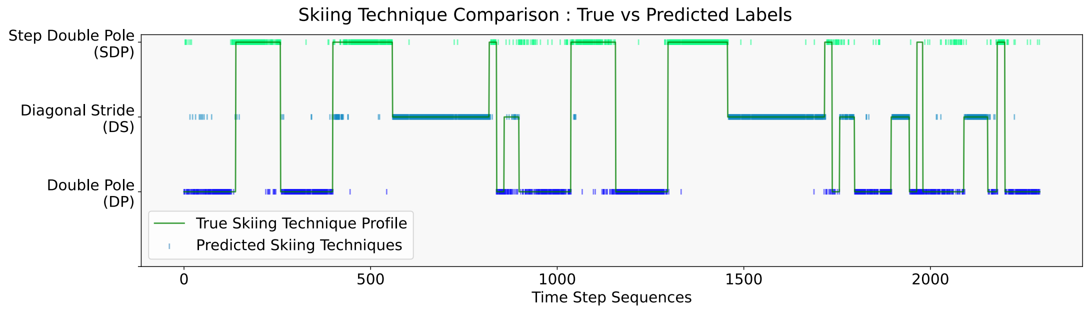
Table 4.8: Cross-validation summary for LSTM Model. Table depicts average across folds, with standard deviation in brackets.

	Model Results Only	Model Results Passed Through Single Filter	Model Results Passed Through Multiple Filters
Accuracy	0.862 (0.019)	0.908 (0.0170)	0.894 (0.022)
Balanced Accuracy	0.856 (0.027)	0.901 (0.028)	0.880 (0.034)
MCC Score	0.785 (0.036)	0.856 (0.031)	0.836 (0.035)

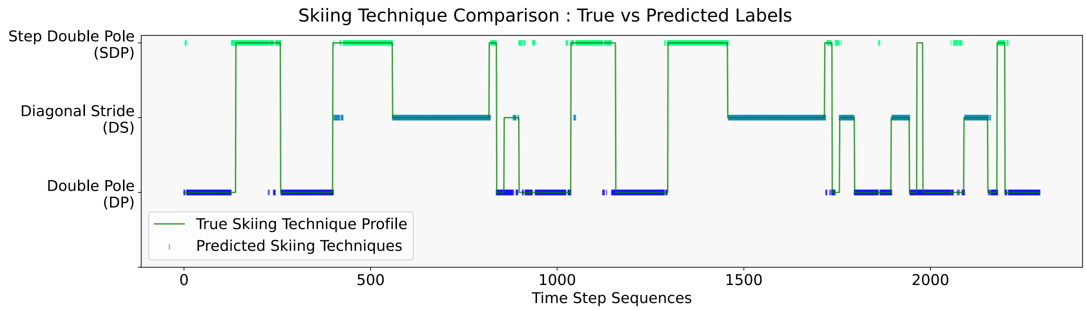
Visualizing model performance

Confusion matrices (Fig. 4.8) were used to assess the classification performance between skiing techniques. Comparison plots (Fig. 4.7) were used to visualize the profile/order the skiing techniques performed and superimpose the predicted labels to see how well the classification performed. Similar to the previous experiment, the folds that performed nearest to the average value are visualized below. Similar to the random forest with reduced feature set, in Fig. 4.8 the classification of accuracy Double Pole and Diagonal Stride skiing techniques was predicted most accurately. Here too, the model was confused the most often between DP and SDP, with SDP being predicted least accurately. This can be expected considering the similarity in the style of skiing between both these techniques where they both comprise of a component of double pole push. The diagonal stride is performed differently compared to these two techniques and thus is confused the least with DP and SDP.

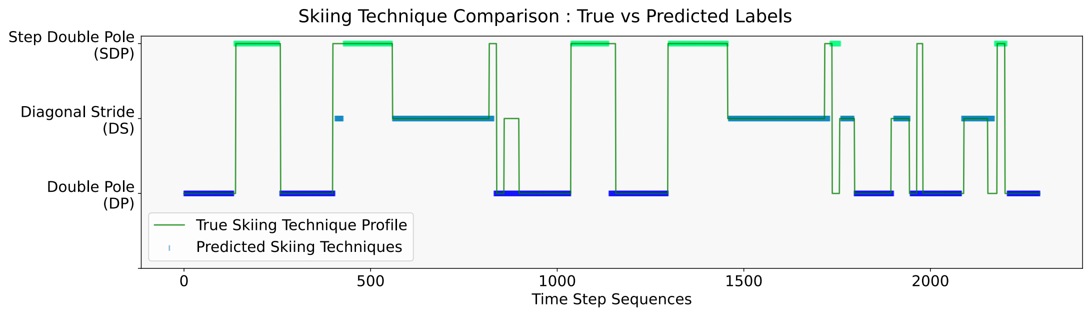
The comparison of true skiing techniques and corresponding predictions are given in Fig. 4.7. The predictions from the model are depicted in Fig. 4.7a and passing the predicted labels through the single filter yields a cleaner grouping of skiing techniques as visible in Fig. 4.7b. The model was able to identify even the smaller duration of DS strokes towards the end of the plots.



(a) Model Results Only

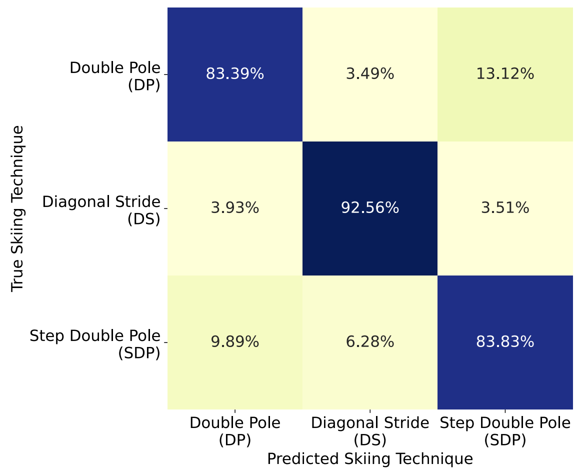


(b) Model Results Passed Through Single Filter

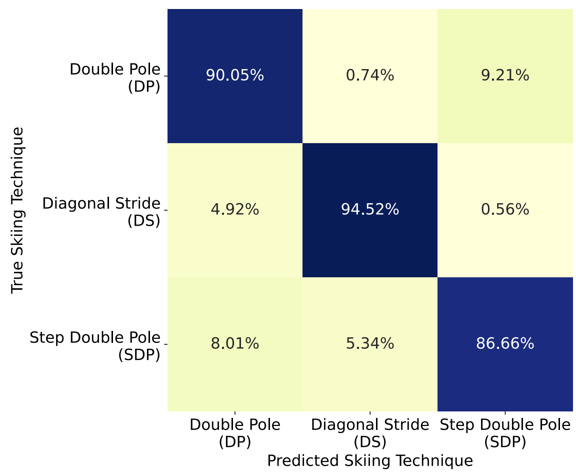


(c) Model Results Passed Through Multiple Filters

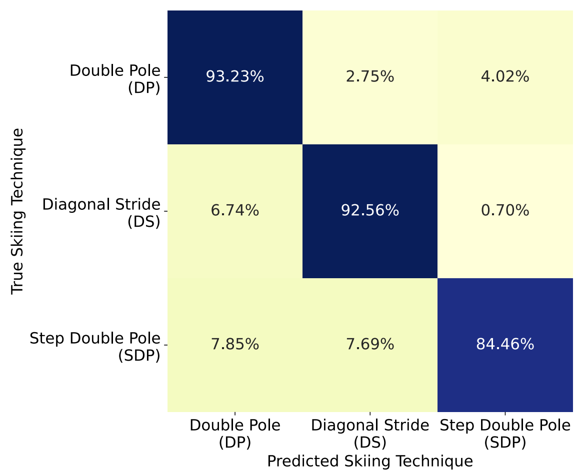
Figure 4.7: Comparison of true skiing techniques vs. predicted skiing techniques



(a) Model Results Only



(b) Model Results Passed Through Single Filter



(c) Model Results Passed Through Multiple Filters

Figure 4.8: Confusion matrix for accuracy of LSTM model

Assessing Model Performance for Treadmill and Outdoor Data

Table 4.9 summarizes the results from just the data that was collected on the treadmill. The model performed similarly to the overall performance, yielding 86.5% accuracy, 86.6% balanced accuracy and a high MCC score of 0.8. There was a low standard deviation of ~2-3%. Passing the predicted labels through a single or multiple majority filter, resulted in a significant improvement of ~5% in accuracy and balanced accuracy. The MCC score increased by 0.07.

Table 4.10 summarizes the results from just the data that was collected outdoors, with 84.1% accuracy but lower balanced accuracy of 74.9%, implying skewed accuracy between classes. It had a higher standard deviation of ~4-5% across overall and balanced accuracy, but a higher deviation of 8% in MCC Score. When the results were passed through the single majority filter. accuracy improved by 3% and balanced accuracy by 2%. However, passing the results through multiple majority filters performed poorly with the performance dropping to 81.9% accuracy and 57.4% balanced accuracy. From Tables 4.9, 4.10 and 4.8 we can see that the LSTM model performed similarly across all datasets, with a minor dip in predictive performance and increase in variability when dealing with outdoor data only.

Table 4.9: Cross-validation summary for LSTM Model on Treadmill Test Data. Table depicts average across folds, with standard deviation in brackets.

	Model Results Only	Model Results Passed Through Single Filter	Model Results Passed Through Multiple Filters
Accuracy	0.865 (0.022)	0.916 (0.023)	0.917 (0.025)
Balanced Accuracy	0.866 (0.022)	0.916 (0.023)	0.917 (0.025)
MCC Score	0.800 (0.032)	0.875 (0.033)	0.878 (0.035)

Table 4.10: Cross-validation summary for LSTM Model on Outdoor Test Data. Table depicts average across folds, with standard deviation in brackets.

	Model Results Only	Model Results Passed Through Single Filter	Model Results Passed Through Multiple Filters
Accuracy	0.841 (0.046)	0.874 (0.057)	0.819 (0.082)
Balanced Accuracy	0.749 (0.058)	0.777 (0.081)	0.574 (0.058)
MCC Score	0.647 (0.085)	0.725 (0.107)	0.570 (0.139)

5

Conclusion

5.1 Discussion

Both the machine learning methods outperform the baseline model. Overall, the LSTM model performs better (accuracy $\uparrow$ by 11.7%) than the random forest model with the reduced feature set. There was low variability in the evaluation metrics ($\downarrow$ by 2.5%) across multiple folds and a significant increase in the MCC Score ($\uparrow$ by 0.187). The LSTM model was able to reproduce high accuracy results more consistently across skiing techniques, highlighted by the higher balanced accuracy of 85.6%. The LSTM model also performed equally well on treadmill and outdoor data and their performance was much closer to the overall performance of the model. Thus, the LSTM model would be able to generalize well if supplied with data collected at any skiing location. The LSTM model performed as well as the random forest with reduced features on outdoor data, but the LSTM edges the random forest in terms of balanced accuracy. The random forest model with the reduced features was definitely the better of the two random forest models as it eliminated the extra features that seemed to be adding noise to the model.

Comparing the confusion matrices of the reduced random forest and the LSTM model, similar patterns were observed. The DP and SDP techniques are similar in hand movements and thus were most frequently confused. SDP was the least accurately predicted technique in the random forests and impacted the overall performance of the models. As expected, DS was easily distinguishable from DP and SDP considering the difference in movements between DS and DP,SDP. DP was the most prevalent skiing technique across the dataset and thus many strokes were incorrectly classified as DP. By extension, when the filtering techniques were applied, DP being the most frequently predicted class, ended up impacting the majority filters causing some of the correct classification to be incorrectly classified as well, for the sake of grouping the close labels together.

Looking at the filtering techniques used, the use of a single majority filter consistently yielded higher performance than the multiple majority filter. However, the multiple majority filter was better at grouping the techniques together as it had multiple passes of varying window length. This cost the model some accuracy, and resulted in smaller groups of labels being re-classified into the dominant class. This can probably be attributed to the basic nature of the design of the filter that could be improved in the future by smarter grouping methods such as distance based clustering or other filters that could yield better results. But, looking at the distribution of the predicted labels it was evident that such an application could make use of an additional refinement step that can improve the interpretation of the results.

Both the machine learning methods performed excellently on data collected on the treadmill. This could be due to the highly controlled environment in which the data was collected. It contained equal duration of each skiing technique and was more likely to be labeled more accurately than the outdoor data as the skiers were following a time-bound test script to switch between the skiing techniques. However, it is also interesting to note that even though there were 11 different skiers that collected data on the treadmill, the models were able to predict the techniques very accurately, potentially across skier attributes such as height, weight and gender. The random forest methods were slightly better than LSTM model in predicting the treadmill data, but had a lower accuracy overall.

5.2 Related Work

An LSTM based approach by Jang et al. [10] suggested 5 sensors as the ideal balance between low number of sensors and a high mean accuracy of 80% for the classical style. When using one sensor on the hip, they observed the overall mean performance dropped to 44.10%. In comparison, with efficient positioning of sensors, such as two sensors in the pole handles, it was possible for us to achieve a high mean accuracy of 86.2%. It highlights the importance of positioning the sensors in ideal locations such that measurements can be taken to help distinguish between skiing techniques. Sandbakk et al. [12] worked with classical style of skiing and tried to improve accuracy of a model by separately labeling the phases of time where a skier is switching between two skiing techniques so that they can be predicted as separate classes. Separating the transitions during prediction and then integrating the results of those transition phases into the skiing techniques yielded an accuracy >90%. In contrast, our study did not separate transition phases from the actual techniques, but instead used the filtering methods to re-label incorrectly classified techniques across the entire skiing duration. Overall, our LSTM model was able to achieve its best overall accuracy of 90.8% with a balanced accuracy of 90.1% implying great performance across all the classes. Even with a lesser efficient multiple filter method, the model achieved an accuracy of 89.4%.

On the other hand, the pilot study with Skisens [13] had a similar setup as our study, but with the machine learning models predicting freestyle skiing technique as opposed to the classical style of skiing we considered. Their study analyzed the time series by extracting skiing strokes of fixed length and padding the beginning and end of the stroke for a shorter stroke duration. For our study, batches of fixed length were considered, however, the time-series was divided without padding every individual stroke. We collected data from a larger number of skiers (14) in both indoor and outdoor scenarios as opposed to only treadmill data. Their study yielded an accuracy of 78% when tested on data from an unseen skier, and we achieved an accuracy of 86.2% on unseen data. Our LSTM model also achieved a mean accuracy of 86.5% on the unseen treadmill data, even when all treadmill skiers were unique with varying heights and weights, thus being able to handle unseen data well.

It is important to note that we were able to achieve comparable results to previous works we surveyed, but with a larger dataset, lesser sensors and lesser input data i.e., only two force measurements. There is also a lot of potential in this approach of skiing technique prediction that uses just two sensors located inside the ski pole handles, as compared to having multiple sensors mounted on the body and equipment of the skiers. These ski pole handles enable zero setup time and a skier is ready to go without having to attach sensors separately. Combining such versatility with a suitable machine learning method could enable skiers to measure and evaluate performance easily and effectively.

5.3 Conclusion

In conclusion, we can say that we were able to fulfill the objectives of the thesis we outlined in the beginning. We have seen that it is possible to achieve high accuracy to predict classical skiing techniques using machine learning models that make use of just two force measurements as input. Both the methods have shown promise in predicting skiing techniques. LSTM Neural networks can be trained with a few more improvements but are usually harder to tune and consume a lot of time and resources to get right. Random forest methods have had limited application in gait analysis, but with the right features, it is possible to develop a machine learning model to address such problems. Random forests are interpretable and have hyperparameters that can be tuned relatively easily. But with larger random forests, interpretability can be difficult when trying to analyze specific scenarios. However, with new tools such as those available in TensorFlow decision forests[1], it can be possible to investigate behavior of random forests in greater detail.

We also saw that the LSTM model can generalize well, predicting treadmill and outdoors skiing data equally well. The random forest performed exceptionally for treadmill data but decreased in performance and had high variability when handling outdoor data. The LSTM model would be our recommendation for future work to fine-tune the existing setup and potentially integrate into applications.

We can confirm that it is useful to create calculated features that could help understand the data and predictions more effectively. Looking at the feature importance plot, we were able to confirm the initial assumptions of the baseline model where we distinguished between techniques using ground contact duration, airtime duration, area under curve and stroke frequency. These features were identified by the random forest model as the most important features to predict the skiing techniques as well. The calculated features are helpful in understanding the nuances of

[1] `https://www.tensorflow.org/decision_forests`

the skiing techniques such as observing the similarities between DP, SDP and dissimilarities between DS and DP,SDP. It helped shed some light on what to expect when performing predictions using the machine learning methods.

Lastly, we saw the benefit of employing filtering methods to refine and group prediction results in such a way that results visually match real-world skiing scenarios better. There was a consistent improvement in prediction when a single majority filter was applied across all machine learning methods. With the help of a simple filter, we may potentially not need additional labels, features to predict transitions.

The structured and unstructured data that has been collected for the purposes of these experiments can be used in the future to build upon the findings from here. It can become part of a larger dataset that potentially comprises of more classical techniques and freestyle data so that models can be trained to cover more cross-country skiing techniques. Data could also be collected in the future for one technique at a time and help train models using the same for applications such as training and calibration to a skiers style of skiing.

Having such predictive power in the form of an application that is used by casual and elite skiers could be beneficial in evaluating performance. The LSTM model can be used as a pre-trained neural network for transfer learning and further improve the performance of prediction. This data could also be used for biomechanical analysis in skiing and identify any other patterns that can differentiate between techniques and thus help in better predictions as well.

Bibliography

[1] Chandra Prakash, Rajesh Kumar, and Namita Mittal. Recent developments in human gait research: parameters, approaches, applications, machine learning techniques, datasets and challenges. *Artificial Intelligence Review*, 49(1):1–40, Jan 2018.

[2] Abdul Saboor, Triin Kask, Alar Kuusik, Muhammad Mahtab Alam, Yannick Le Moullec, Imran Khan Niazi, Ahmed Zoha, and Rizwan Ahmad. Latest research trends in gait analysis using wearable sensors and machine learning: A systematic review. *IEEE Access*, 8:167830–167864, 2020.

[3] Huanghe Zhang, Yi Guo, and Damiano Zanotto. Accurate ambulatory gait analysis in walking and running using machine learning models. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 28(1):191–202, 2020.

[4] Johannes Tjønnås, Trine M. Seeberg, Ole Marius Hoel Rindal, Pål Haugnes, and Øyvind Sandbakk. Assessment of basic motions and technique identification in classical cross-country skiing. *Frontiers in Psychology*, 10:1260, 2019.

[5] Valentina Camomilla, Elena Bergamini, Silvia Fantozzi, and Giuseppe Vannozzi. Trends supporting the in-field use of wearable inertial sensors for sport performance evaluation: A systematic review. *Sensors*, 18(3), 2018.

[6] Kevin King, Jessandra Hough, Ryan McGinnis, and N. Perkins. A new technology for resolving the dynamics of a swinging bat. *Sports Engineering*, 15, 03 2012.

[7] Sacha K. Fulton, David B. Pyne, and Brendan Burkett. Validity and reliability of kick count and rate in freestyle using inertial sensor technology. *Journal of Sports Sciences*, 27(10):1051–1058, 2009. PMID: 19642049.

[8] Finn Marsland, Keith Lyons, Judith Anson, Gordon Waddington, Colin Macintosh, and Dale Chapman. Identification of cross-country skiing movement patterns using micro-sensors. *Sensors*, 12(4):5047–5066, 2012.

[9] H. Myklebust, T. Losnegard, and J. Hallén. Differences in v1 and v2 ski skating techniques described by accelerometers. *Scandinavian Journal of Medicine & Science in Sports*, 24(6):882–893, 2014.

[10] Jihyeok Jang, Ankit Ankit, Jinhyeok Kim, Young Jae Jang, Hye Young Kim, Jin Hae Kim, and Shuping Xiong. A unified deep-learning model for classifying the cross-country skiing techniques using wearable gyroscope sensors. *Sensors*, 18(11):3819, November 2018.

[11] Thomas Stöggl, Anders Holst, Arndt Jonasson, Erik Andersson, Tobias Wunsch, Christer Norström, and Hans-Christer Holmberg. Automatic classification of the sub-techniques (gears) used in cross-country ski skating employing a

mobile phone. *Sensors (Basel, Switzerland)*, 14(11):20589–20601, Oct 2014. 25365459[pmid].

[12] Ole Rindal, Trine Seeberg, Johannes Tjønnås, Pål Haugnes, and Øyvind Sandbakk. Automatic classification of sub-techniques in classical cross-country skiing using a machine learning algorithm on micro-sensor data. *Sensors*, 18(2):75, December 2017.

[13] Moa Johansson, Marie Korneliusson, and Nickey Lizbat Lawrence. Identifying cross country skiing techniques using power meters in ski poles. In Kerstin Bach and Massimiliano Ruocco, editors, *Nordic Artificial Intelligence Research and Development*, pages 52–57, Cham, 2019. Springer International Publishing.

[14] Elham Rastegari, Sasan Azizian, and Hesham Ali. Machine learning and similarity network approaches to support automatic classification of parkinson's diseases using accelerometer-based gait analysis. In *Machine Learning and Similarity Network Approaches to Support Automatic Classification of Parkinson's Diseases Using Accelerometer-based Gait Analysis*, 01 2019.

[15] Ayushi Gupta, Apoorva Jadhav, Sanika Jadhav, and Anita Thengade. *Human Gait Analysis Based on Decision Tree, Random Forest and KNN Algorithms*, pages 283–289. ., 01 2020.

[16] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.

[17] Ralf C Staudemeyer and Eric Rothstein Morris. Understanding lstm–a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*, 2019.

A

Appendix

A.1 Data Collection : Treadmill Protocol

Pre-defined protocol for data collection on the treadmill.

Table A.1: Treadmill Protocol

Time (min)	Slope (%)	Speed (km/h)	Technique
0	2	14	Double Pole
0.5	2	16	Double Pole
1	2	18	Double Pole
1.5	2	20	Double Pole
2	2	22	Double Pole
2.5	2	24	Double Pole
3	2	14	Double Pole
3.5	4	10	Step Double Pole
4	4	12	Step Double Pole
4.5	4	14	Step Double Pole
5	4	16	Step Double Pole
5.5	4	18	Step Double Pole
6	4	20	Step Double Pole
6.5	4	18	Double Pole
7	4	16	Double Pole
7.5	4	14	Double Pole
8	5	14	Double Pole
8.5	6	14	Double Pole
9	7	14	Double Pole
9.5	3	14	Double Pole
10	3	12	Step Double Pole
10.5	4	12	Step Double Pole
11	5	12	Step Double Pole
11.5	6	12	Step Double Pole
12	7	12	Step Double Pole
12.5	8	12	Step Double Pole
13	9	10	Step Double Pole
13.5	10	10	Step Double Pole
14	6	8	Diagonal Stride

14.5	8	8	Diagonal Stride
15	10	8	Diagonal Stride
15.5	12	8	Diagonal Stride
16	14	8	Diagonal Stride
16.5	16	8	Diagonal Stride
17	6	9	Diagonal Stride
17.5	6	10	Diagonal Stride
18	7	11	Diagonal Stride
18.5	8	12	Diagonal Stride
19	9	13	Diagonal Stride
19.5	8	14	Diagonal Stride
20	7	15	Diagonal Stride
20.5	7	15	Step Double Pole
21	7	15	Double Pole
21.5	7	11	Diagonal Stride
22	6	10	Diagonal Stride
22.5	6	8	Diagonal Stride

A.2 Data Collection : Description of CSV file

Table A.2: Description of fields recorded in the csv files

Field Name	Field Description
t_prot (min)	Timestamp recorded on the treadmill
t_left (min)	Timestamp recorded on the left pole sensor
t_right(min)	Timestamp recorded on the right pole sensor
f_left (N)	Force measured on the left pole sensor
f_right (N)	Force measured on the right pole sensor
slope (%)	Inclination of the treadmill
speed (km/h)	Speed of the treadmill
gear	Skiing technique performed