An AI Prediction Model for the PGA Tour

Josh Beal
gx20774@bristol.ac.uk

Ben Hoddinott
rf21387@bristol.ac.uk

Archie Green
fu21381@bristol.ac.uk

Charlie Gardiner
uk21685@bristol.ac.uk

***Abstract*—This report applies deep learning techniques for predicting golf scores and tournament outcomes. The study explores the efficacy of various models, including a baseline linear regression model, a Feedforward Neural Network (FNN), and Recurrent Neural Networks (RNNs).**

Using a comprehensive dataset sourced from the PGA Tour and DataGolf, the models are trained and evaluated on historical player statistics. The evaluation on testing data indicates that the FNN model performs the most accurately with an RMSE of 0.699, followed by linear regression, which achieved RMSE of 2.165, while the LSTM and GRU models had RMSE's of 2.804 and 3.226 respectively. The neural network models had considerably larger computation times than the baseline model, by 3 orders of magnitude.

Also, the paper calculates the win percentage for golfers in tournaments by accounting for the difficulty of the course as well as player score and current form. The simulation of multiple scenarios and consideration of performance variability, provides a probabilistic forecast for tournament outcomes.

This study highlights the potential of AI-driven approaches in enhancing sports analytics and informing strategic decision-making. By presenting rigorous experimentation and analysis, the paper highlights the predictive capabilities of deep learning models in golf forecasting.

I. Introduction

Golf saw a huge resurgence of popularity during 2020 [1], due to it being one of the few activities to open up early during the COVID-19 pandemic. Golf, being a very inclusive sport, can be enjoyed by people of all ages. This increased participation and brought attention to the game, including viewership and interaction with the professional game. The 2023 PGA Tour saw a 31% increase in social media engagement from that of 2022 [2]. This increased interest sparked development in and around the game.

For the average professional golfer, there is a minimum cost of around $\$180,000 - \$200,000$ to compete every year on the PGA Tour [3]. During a four day tournament, the bottom half of golfers will miss the 'cut', meaning they don't compete in the last two days of the competition. In the Players' Championship, there is a total prize pool of $25 million, however, if a player fails to make the cut, they get no prize money which is the same for all PGA Tour events [4]. Every golfer is responsible for travel expenses and caddy fees, meaning it can be risky for lower-rated players to go to every tournament when there is no guarantee of prize money.

Below the PGA Tour, there is the Korn Ferry Tour, where the total prize money is around 6% that of the PGA Tour whilst expenses stay at around 40% [5][6]. The endorsements for the Korn Ferry are very sparse in comparison, especially because it is not televised. As a result, each player has to make the decision of whether or not to compete in a tournament from a financial standpoint. A model that can give these players further insight into their chances of finishing in a top percentage of the field would aid in making this decision.

Golf is an extremely challenging sport to predict as there are many factors at play during a round of golf. This report aims to predict golfers scores and therefore the outcome of certain tournaments and matches using deep learning.

An initial model using simple linear regression to predict round scores as a baseline model will allow us to see how accurate the other models are. A feed-forward neural network that uses regression and two separate Recurrent Neural Networks (RNN) models will also be used. One of these uses a Gated Recurrent Unit (GRU) whilst the other uses a Long Short-Term Memory (LSTM) unit.

II. Literature Review

Models predicting the outcome of sports events are increasingly common over a range of sports. This literature review will focus on a previous golf prediction method, prediction methods for horse racing and a discussion of RNNs.

The website 'DataGolf' has a model to predict the percentage chances of a golfer winning a tournament [7]. Their model uses ordinary least squares (OLS) linear regression to predict golf round scores. There are many benefits and limitations to OLS. It is a simple method, making it easy to interpret, but this also means it can't capture complex, non-linear relationships within the data. OLS assumes a linear relationship between the independent and dependent variables. If this assumption is broken, the model's predictions may be biased and inaccurate. Therefore, using a more complex model such as a neural network might be better as it can capture more complex, non-linear relationships.

Also, DataGolf's method for calculating the win percentage of a golfer for a tournament includes taking all of the predicted scores for each golfer and adding a random, normally distributed number to their predicted score. Scores are then ranked from lowest to highest. This process is repeated 1500 times and the proportion of times a golfer wins is used to calculate their percentage win rate. Each golfer has their own normal distribution with a specific variance but everyone has the same mean, which is equal to zero. This player-specific variance is the variance of their previous round scores, known as consistency. It is calculated by taking the residuals from previous rounds and estimating the sample variance of these residuals. This is a good method to calculate the win rate for each golfer, but this is limited by the variation in difficulty of every golf course. Therefore, normalising a player's score by the course difficulty would more accurately calculate their consistency leading to a more accurate win rate. The PGA rank the difficulty of the courses it plays and also gives the average score as a number which can be used to normalise a player's score[8].

A second model uses neural networks to predict the outcomes of horse races[9]. Horse racing is a similar sport to golf because they are individual sports where each person or horse is ranked from first to last. Another similarity is the amount of unpredictability in both sports, for example, a horse falling at a fence or a golfer chipping in from 150 yards. In this model, they train a neural network for each horse and output a time that horse will finish the race in. The prediction is then based on each horse's time to finish the race. This network structure involves training a regressor instead of a classifier. However, this model does not consider the interactions between horses implying that each horse is independent of one another - an assumption which is not accurate. This paper's results show that simple backpropagation (backprop) is the most effective algorithm when compared to four others, backprop with momentum, Quasi-Newton, Levenberg-Marquardt and conjugate gradient descent. However, it does take the longest to train, with Levenberg-Marquardt being the fastest. The structure of the neural network is 8-5-7-1, where 8 is the number of input features.

RNNs are another broad type of artificial neural network which conserves information between layers. In essence, a node in a previous layer can affect the activation in a subsequent layer. Their ability to hold and distribute information in this way makes them ideal for many applications including sequence prediction and speech recognition tasks [10].

The hypothesis that golf is a form based sport where recent high or low scoring rounds can influence a player's next score is proposed. To test this, a variety of RNNs can be implemented and compared to give an understanding into the effectiveness of 'past performance' based prediction in golf.

LSTM networks are a type of RNN which were invented in 1997 by Hochreiter and Schmidhuber [11], but came into prominence in 2007 where they have been applied to many problems. They are unique due to their ability to take into account both long and short term trends, which makes them particularly useful for sequence prediction tasks. GRU networks were developed in 2014 by Kyunghyun Cho et al [12] and share many similarities with LSTM networks including it's ability to handle time series dependency within data.

III. Methods

A. Data Collection and Pre-processing

For this model, the PGA Tour was chosen due to it's extensive, in-depth datasets as well as it being widely regarded as one of the most popular professional tours in the world. The PGA Tour website contains years of historical data from every PGA tournament [13]. However, the DataGolf API [14], allows for efficient extraction of this data from the PGA website. The number of features used in our models to predict a player's score is limited because only a certain amount of recorded statistics are available. Using an API key the data acquired includes the number of shots gained putting, shots gained around the green, shots gained off the tee, shot gained from tee to green, total shots gained, driving distance and driving accuracy. As well as these statistics, player name, course name and the date of play are extracted. The course where the round was scored is also presented as a non-numeric variable. Due to the inability for neural networks to process string inputs, one-hot encoding is used for the course names within the dataset, enabling the neural network to use it as an input feature. This data is extracted for all PGA golfer's previous performances. Each golfer's current statistical form is also given by DataGolf, which is a 10 round rolling average of their most recent playing statistics. This can be used as an input to certain models to predict future rounds.

The data is split into three sets - train, validation and test. The training set is used for implementing the back-propagation algorithm, while the validation set helps to identify over fitting and helps tune a model's hyper-parameters. The test set then gives insights into the models capability to generalise to unseen data. The input features of the model are scaled. This helps speed up the convergence to the minimum of the loss function but more importantly, each feature has equal importance. Without scaling, features with larger values could disproportionately influence the model's learning.

B. Baseline Model

It is important to have a baseline model to compare the other models to in order to evaluate their relative performance. In this case there are a limited number of open source models for predicting golf scores. Therefore, a baseline model will be developed using the exact same data that the other more complex models are trained on. This is vital to compare the models accurately, as it provides a reference point that allows us to measure any incremental benefits.

A linear regression model is chosen as the baseline model. This is because linear regression is simple and easy-to-understand and is unlikely to overfit due to the large number of observations in the dataset (87,000 rounds of golf). Also, it is very computationally efficient, meaning it can be done quickly and efficiently. This model is fitted to all available data extracted.

C. Feedforward Neural Network (FNN)

Building upon the simple regression model a fully connected, feed-forward, neural network is implemented and adapted to complete regression-based tasks. Regression is chosen over classification because, in classification, an incorrect categorisation is evaluated as wrong. However, regression allows for predictions that are close to the actual value. Therefore, in scenarios like predicting a golf score, being closer to the actual score is better than being classified incorrectly, making regression the better choice for predicting golf scores.

The input layer consists of neurons corresponding to the input features which are a player's current form statistics.

The network outputs a single continuous value as a round prediction. The network uses an Adam optimiser as it is considered best practice [15]. The network is given 2 hidden layers, as seen in previous literature section II. Rectified Linear Unit (ReLU) is used as the activation function in this model as it is widely seen to have better performance in training than its 'sigmoid' counterpart [16]. Mean squared error (MSE) is used as the loss function, in order to get accurate predictions.

D. Recurrent Neural Networks (RNN)

Unlike the FNN model, RNNs do not make any assumptions about a player's future round statistics which are unknown at the time of prediction. Instead RNNs will take into account historical data and trends to make predictions. Removing this assumption, which is a limitation to the FNN, makes RNNs more applicable and realistic.

An RNN will be trained for each player as each will have unique historical trends and data. A player's consistency, or the way they bounce back from potential slumps in form, represents important data points for these model types to understand, as they are characteristics based on the individual. Therefore, the data will be grouped by player and sorted in chronological order to understand these temporal trends.

Once a dataset for a player has been extracted in temporal order it is split into sequences. These sequences are created using a 'sliding window' approach, where each sequence consists of a fixed number of consecutive data points. The number of data points or sequence length is interpreted as the number of previous rounds taken into account when making a prediction.

1) Long-Short-Term Memory (LSTM): LSTM architecture contains a unit with 3 gates (forget, input and output) which control the information passed through the network, enabling dependencies to be captured. This specific network architecture is trained to a single player's data using mean squared error (MSE) as the loss function which aims to give accurate continuous golf round predictions. The Adam optimisation algorithm is used to train the network using the training data, which is in the form of sequences.

2) Gated Recurrent Unit (GRU): GRU architecture is simpler and only contains a single update gate which controls information transfer. As such this network has fewer parameters than an LSTM unit. This often makes them faster to train and less prone to over-fitting. This GRU structure is implemented in the same way as the LSTM structure using an MSE loss function and the Adam optimiser.

E. Performance Predictor

A key part of this model is to evaluate the performance for each player at future tournaments. This performance predictor calculates a player's percentage chance of reaching a desired result, be it winning, coming top 10 or making the cut. This is critical as it gives an insight into how a golfer might perform at an event, whilst introducing an element of unpredictability which is often seen in golf.

Expanding on work reviewed in section II, Equation 1 creates a Normalised Player Par Score (PPS_N) by subtracting the Course Average Par Score ($CAPS$) from the Player Par Score (PPS). PPS is defined as the number of shots over or under a player is from the course par.

$$PPS_N = PPS - CAPS \tag{1}$$

Therefore this PPS_N takes into account the difficulty of a course, giving a more accurate method of recording a player's performance for a round. Then the standard deviation for a player's normalised scores is calculated, which represents how consistently they have played throughout their career.

A normal distribution is then fitted to each player using their specific standard deviation and a mean of zero. A mean of zero is selected as this number is to be added to a golfer's predicted score for a round.

A number is then randomly sampled from each player's distribution and added to their predicted round score to form the final rankings for a round. This process is repeated 1500 times to increase accuracy, obtaining 1500 potentially different results. Then the players percentage chance of a desired result (PC) can be calculated using Equation 2.

$$PC = \frac{\text{No. Times}}{1500} \times 100, \tag{2}$$

Where $No.Times$ is the number of times the player achieves the desired result. For example, for win percentage it would be the number of times the player has the lowest score after their unique normal distribution sample is added.

IV. Experimentation

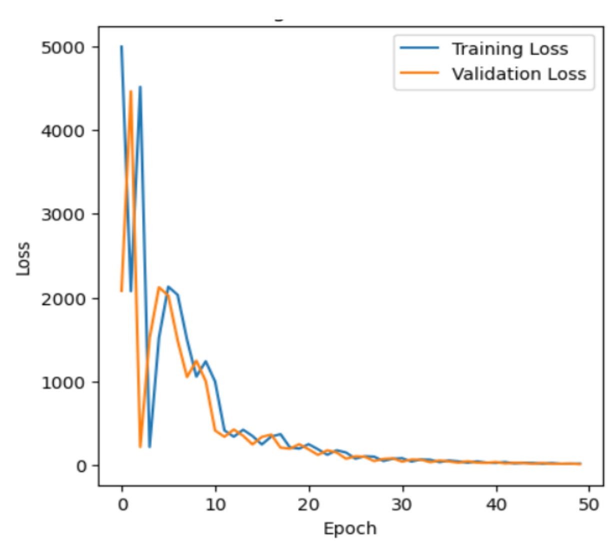

Fig. 1. Number of epochs against the validation and training loss for the FNN model.

Figure 1 shows the FNN training and validation loss curves which are similar for all three neural network models, each differing slightly. In Figure 1 the loss starts high at a small number of epochs, as expected, and falls sharply showing the model's ability to learn trends within the data. The loss then rises but sharply falls again by 10 epochs. Consequently, as the learning rate is reduced, the loss oscillates closer to the minimum, resulting in slower but more stable convergence towards the minimum of the loss function. After 100 epochs the reduction of loss is minimal for the FNN model while the computation time rises significantly. Therefore, 100 epochs are used for the FNN model and for the same reasons, 200 epochs are used for the LSTM and GRU models. Figure 1 also shows that the validation and training loss decrease proportionally, showing the model does not overfit and should generalise well to unseen data, again seen with the LSTM and GRU models. Through experimentation with the learning rate for the FNN it was found that 0.01 was the optimal as it is the largest value that converged, thus reducing computation time.

The number of golf rounds or sequence length in the training of both GRU and LSTM models is an unknown parameter. By training and testing each model independently on different sequence lengths, a better understanding of an optimal sequence length can be obtained. It is important to note that each player will have different optimal sequence lengths for both models, where some players' scores may be more suited to LSTM or GRU predictions.

To evaluate each sequence length's performance, the RMSE on the test set could be evaluated. However a more current and topical result would involve testing each model's prediction accuracy in an upcoming tournament. Here Rory McIlroy's next round score is predicted for the Arnold Palmer Invitational 2024 at Bay Hill Club & Lodge. Figure 2 shows 10 independent models trained for each sequence length with no fixed random seed to assess the stability and variability of the model's performance across different runs (Monte Carlo Experiments). In each case, the model predicts McIlroy's next tournament score at Bay Hill Club & Lodge. This process is applied to all predicted golfers.

The experiments visualised in Figure 2 show that the LSTM model gives more stable predictions for larger sequences but predicts more inconsistently for smaller sequences. The model tends to take into account longer term trends and is seen to just predict McIlroy's average score for longer sequences which does not provide specific insights into granular performance. However, for shorter sequences, the model can be interpreted to be more variable with its predictions and has less information leading to a rise in prediction variance.

Overfitting occurs when the model is learning the mean score. This is useless in terms of prediction based models, so a trade-off must be found between overfitting and being variable in prediction. Figure 2 shows that a sequence length of 4 gives the lowest mean error to McIlroy's actual tournament round score showing it's ability to predict accurately. Predictions at this sequence length also show some level of variance exhibiting the model's tendency to vary from the mean, making this sequence length optimal.

Figure 2 shows the GRU has results that are less patterned than that of the LSTM predictions. On the other hand this model seems to predict more accurately on average than that of

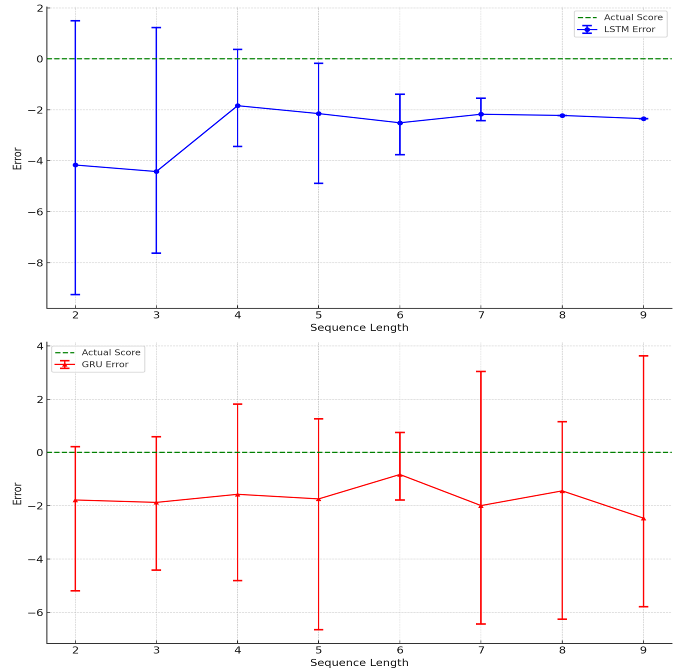

Fig. 2. The graph presents the evaluation of 10 independently trained models for each sequence length, spanning sequence lengths from 2 to 9. It visualises the average error and the range of errors encountered when trained on Rory McIlroy's dataset.

the LSTM network. The mean error for each sequence length tends to be smaller but with a greater prediction variance, particularly when using longer sequences. A GRU network does not take long term trends into account like an LSTM network and also has a simpler structure which can lead to more inconsistent predictions, as seen in Figure 2. However, specifically for McIlroy's dataset a sequence length of 6 appears optimal. For McIlroy the GRU network in general is more accurate at predicting.

Whilst experimenting with different learning rates for RNN models, it is found that very small learning rates converge to a minimum of the loss function over many epochs. For both RNN models, in similar ways to sequence length, the lower learning rates converged to a player's average score. This is due to the model being trained extensively through an MSE loss function. This theoretically is a loss which needs to be minimised but can lead to models that can overfit to the data. By overfitting in this way the models miss nuanced patterns or specific variations to certain observations - an important feature of RNNs. For the models to make useful predictions to unseen data in future predictive tasks, each RNN model is trained with a learning rate of 0.1. Using lower values for this hyper-parameter is seen to make the model overfit.

Another experimental method considered was early stopping in order to train the model over less epochs. This regularisation technique was implemented but was found to give more unstable results.

Table I shows the average test set RMSE, memory usage and computational time, for all players in each model. Optimal sequence length's has been chosen for RNN models.

TABLE I
ROOT MEAN SQUARED ERROR (RMSE) ON TEST DATA, COMPUTATION TIME AND MEMORY USAGE FOR EACH MODEL. THE OPTIMAL RESULTS ARE SHOWN IN **BOLD**.

Model Used	RMSE	Computation Time (s)	Memory Usage (mb)
Linear Regression	2.165	**0.009**	**228.5**
FNN	**0.699**	19.750	1079.50
LSTM	2.804	38.250	733.16
GRU	3.226	4.510	698.46

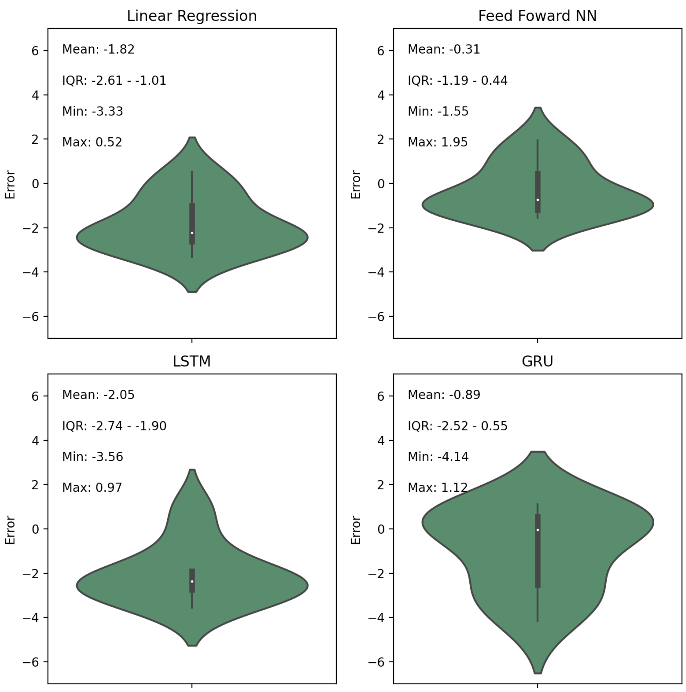

Fig. 3. Violin plots showing the mean, interquartile range, max, min and density of predicted error values for the 10 golfer's tested for each different model type.

The four models explained in section III are each used to predict the average score for 10 players over four rounds at the Arnold Palmer Invitational at Bay Hill Club & Lodge. The difference between each model's predictions and the 10 players actual scores are represented in Figure 3.

Table II gives an example output showing the prediction of the win percentage of each player from the FNN model at the Arnold Palmer Invitational on 7-10 March 2024. The Actual Mean Score is their average score over the four rounds of the tournament, then they are ranked in order of where they actually finished.

TABLE II
TABLE SHOWING THE WIN PERCENTAGE AGAINST THE OTHER 9 PLAYERS TESTED, FOR THE ARNOLD PALMER INVITATIONAL 2024. THIS IS THEN COMPARED WITH THEIR ACTUAL SCORES AND RANKED AMONGST THEMSELVES. T FOLLOWED BY A NUMBER MEANS TIED FOR THAT POSITION.

Name	Win Percentage	Actual Mean Score	Actual Rank
Scheffler	20.67	68.25	1
Clark	7.67	69.50	2
Homa	7.60	71.00	3
McIlroy	7.33	71.75	4
Schauffele	10.67	72.00	T5
Aberg	8.80	72.00	T5
Burns	13.13	72.25	T7
Spieth	12.40	72.25	T7
Cantlay	6.60	72.50	T9
Fowler	5.13	72.50	T9

V. DISCUSSION

RNNs are known to have limitations in their training process when backprop is implemented. One limitation is the exploding or vanishing gradient problem and is a result of training with larger sequences [17]. As sequence length increases the magnitude of the derivatives when propagated back through time reduces. In principle this slows down the training process and can even stop the RNN from learning completely. Conversely, the exploding gradient problem can cause gradients to grow exponentially large during backprop. The specific sequence length where this would occur for this problem is unknown, however it is worth considering if the RNN models are implemented using a greater sequence length than that in Figure 2.

As seen in Table I, the GRU network has a lower computation cost than the other neural network models tested. This is due to the GRU having a more simplistic structure and less parameters to update when implementing the backprop algorithm. Therefore the GRU may be better suited to be trained on larger datasets. The LSTM model shows nearly 8.5 times larger computation time than the GRU but yields a smaller average RMSE when evaluated on all tested golfers. Depending on the dataset used and accuracy requirements, the experimentation in section IV could inform a choice between LSTM and GRU networks.

LSTM networks have more complicated structure and more parameters to tune, which can cause them to overfit especially to smaller datasets. The LSTM model is prone to this as each player's data is not extensive. However, through regularisation techniques explored in section IV, including adjusting sequence length and learning rates, the LSTM model is proven to be the better model on average in making score predictions, as seen in Table I.

Through experimentation it is found that training each RNN with a learning rate less than 0.1 finds the minimum of the MSE loss function. This however was found to just be the average player score. The loss function used here may be sub-optimal. Finding a loss function which accurately predicts round score whilst not defaulting to the mean of a players round scores would provide more useful results. Experimenting with different loss functions including mean absolute error (MAE) or determining a custom loss function could give more interesting results.

The comparison of models shown in Table I suggests that the FNN is the optimal prediction model to use. It's computational cost and memory usage is larger than all other models, but it is chosen due to it's RMSE being the lowest, as it is three times more accurate than the others. The computational times in Table I, although all relatively small, could be a greater factor if the models are used on much larger datasets.

The FNN and linear regression models could have lower RMSE values due to the fact they are trained and tested based on shots gained throughout previous rounds. Where the input to the models includes the golfer's current form, which is based on their 10 most recent rounds.

Limitations arise with the GRU and LSTM models because they only train and predict for a certain golfer's data. This means the accuracy will be reduced as a smaller portion of the dataset has been used in training. It is also shown by the decreased memory usage compared to the FNN model. It is possible the accuracy could be greatly improved with a larger dataset for each golfer. The baseline linear regression model shows its simplicity through its low values for computation time and memory usage being by far the lowest of all models.

Figure 3 reinforces the notion that the FNN model is optimal. It shows that out of the predictive models FNN has the smallest mean error and smallest range of error. The graphs show that each model used under-predicts the scores on average. This could be due to the models not training effectively on course difficulty. Since Bay Hill Club & Lodge is a tough course the golfers take more shots, causing an under-prediction in round score by all models. This aga

Table II shows an example output from the FNN model. FNN is used as it has the best RMSE in Table I, but this win percentage can be applied using the outputs from all models. It produces a player's probabilities, where win percentage is chosen here. Then, these percentages are compared to their actual ranking against the other 9 players. Whilst golf is a difficult sport to predict, emphasised by the difference in percentage of winning from the best to the worst being approximately 15%, it can be seen that Scheffler had the highest chance of winning which was predicted correctly. In a full tournament there are around 140 players. This is where the computational time is important. On a large scale, using data for 140 players for four rounds and then running them through the performance predictor can be a time consuming task. Therefore, whilst the FNN model has the lowest absolute error, using the Linear Regression model may be beneficial for a large scale predictions if used over many different competitions in order to save computational time.

VI. Conclusion

In conclusion, this project presents a comprehensive exploration into predicting golf scores using various neural network models and simple linear regression as a baseline. Using data from PGA Tour events and player-specific statistics, the models were trained to forecast the performance of golfers in a future tournament.

The results demonstrate the effectiveness of different models in predicting golf scores, with the FNN emerging as the most accurate among the tested models. While FNN provides superior accuracy, it also demands higher computational resources compared to the simpler linear regression model, which had the second lowest RMSE. The LSTM and GRU models offer insights into capturing temporal trends in golf performance, which is not seen to be as effective as other models. The hypothesis stated in II can therefore be concluded false.

Furthermore, the report introduces a novel approach to compute each players performance in a tournament, providing valuable insights for both bettors and golfers, as this can be used to predict if golfers will make the cut, make top 10 or win. By simulating multiple scenarios and considering the variability in player performance, the model offers a probabilistic outlook on tournament outcomes.

VII. Future Work

Within neural networks, there are always more parameters to be explored in order to increase the accuracy of the outcome. One route for further developments of this project could be to include weather data in our neural network. By training with weather features such as wind speed, rainfall and ultraviolet (UV) intensity, the network would have more information to predict a golfer's round score. This addition of weather features into the data could allow for more accurate predictions. Additional parameters could include statistics of separate holes, such as how it can be played and the difficulty of each hole.

Another way to further improve our model would be to introduce a live model. As a tournament goes on, the chances of players coming in a top percentage of the field is likely to change as they play the course. Therefore, a live updating model which takes into account the way a player is competing throughout the round compared to the rest of the field would be a good way to develop our model. This project could be improved by training on more course related statistics along side player statistics, allowing a more course emphasis prediction based model.

Finally, this model can be used for player development. The model can be developed to coach players in what to work on in their game. For example, ahead of a tournament, a coach could analyse the course statistics alongside the current form of the player. If the player can improve their putting to a certain standard over a period of time, the stats can then be used in the model to predict where they will then finish in the tournament. Alternatively, they can look at working on their driving and compare their prediction to the improved putting prediction during the same time period.

GITHUB URL: https://github.com/jjb023/AI-Golf

References

[1] Mike Stachura. The numbers are official: Golf's surge in popularity in 2020 was even better than predicted — golfdigest.com. https://www.golfdigest.com/story/national-golf-foundation-reports-numbers-for-2020-were-record-se, 2021. [Accessed 27-02-2024].

[2] Kiko Badillo. How Many People Watch Golf? [Golf Viewership Statistics 2024] — playtoday.co. https://playtoday.co/blog/stats/golf-viewership-statistics/, 2024. [Accessed 27-02-2024].

[3] Alex Myers. Pga tour pro reveals his average weekly expenses, and let's just say you better be making cuts out there. https://www.golfdigest.com/story/pga-tour-pro-reveals-weekly-expenses-ben-griffin, Oct 2023.

[4] https://sports.betmgm.com/en/blog/pga/pga-tour-biggest-purses-prize-money-this-season-bm10/, Mar 2024.

[5] Staff. How it works: 2023 korn ferry tour. https://www.pgatour.com/korn-ferry-tour/article/news/latest/2023/how-it-works-earn-pga-tour-card-structure-changes-eligibility-format-faq, Jan 2023.

[6] Aaron and Christian. How much does it cost to play professional golf? https://www.gutsygolf.com/how-much-does-it-cost-to-play-professional-golf/, Jun 2023.

[7] Predictive model methodology — datagolf.com. https://datagolf.com/predictive-model-methodology/#page-footer9. [Accessed 15-02-2024].

[8] Toughest Courses : PGA TOUR Media Guide — pgatourmediaguide.com. https://www.pgatourmediaguide.com/records/toughest-courses. [Accessed 28-02-2024].

[9] Elnaz Davoodi and Alireza Khanteymoori. Horse racing prediction using artificial neural networks. 06 2010.

[10] What are Recurrent Neural Networks? — IBM — ibm.com. https://www.ibm.com/topics/recurrent-neural-networks. [Accessed 19-03-2024].

[11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[12] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[13] Pga Tour. Scoring: Categories: Pga tour stats. https://www.pgatour.com/stats/scoring.

[14] Data Golf — datagolf.com. https://datagolf.com/. [Accessed 27-02-2024].

[15] Ruslan Abdulkadirov, Pavel Lyakhov, and Nikolay Nagornov. Survey of optimization algorithms in modern neural networks. *Mathematics*, 11(11), 2023.

[16] Michael Biehl Elisa Oostwal, Michiel Straat. Hidden unit specialization in layered neural networks: ReLU vs. sigmoidal activation — sciencedirect.com. https://www.sciencedirect.com/science/article/pii/S0378437120308153, 2021. [Accessed 18-03-2024].

[17] Dr Barak Or. The Exploding and Vanishing Gradients Problem in Time Series — medium.com. https://medium.com/metaor-artificial-intelligence/the-exploding-and-vanishing-gradients-problem-in-time-series-6b87d558d22. [Accessed 19-03-2024].