UFC Fight Prediction using Machine Learning

James Peralta
Department of Computer Science
University of Calgary
Calgary, Canada
james.peralta@ucalgary.ca

Abdulkareem Dolapo
Department of Electrical and Computer Engineering
University of Calgary
Calgary, Canada
abdulkareem.dolapo@ucalgary.ca

Satyaki Ghosh
Department of Software Engineering
University of Calgary
Calgary, Canada
ghoss@ucalgary.ca

Anil Sood
Department of Software Engineering
University of Calgary
Calgary, Canada
anil.sood2@ucalgary.ca

***Abstract*—The Ultimate Fighting Championship (UFC) is a Mixed Martial Arts (MMA) promotion company based in Las Vegas and is currently the largest organization for MMA in the world with more than 2.4 million pay-per view buys for major events. Fans all over the world are tuning in to see who will win between the best MMA fighters in the world. Machine learning techniques have produced state of the art results in predicting the winner of sports events but has rarely been applied to the combat sports industry where the outcome of a fight could be decided by one lucky punch. In this paper we are predicting outcomes of UFC events using both traditional machine learning techniques such as logistic regressors and SVMs, and state of the art techniques such as Dense Neural Networks (DNNs), Convolutional Neural Networks (CNNs) and Long Short Term Memory (LSTMs). Using the combat statistics and history of two fighters we were able to predict the outcome of a fight between them with an accuracy of 75%. We also performed some analysis on a per fighter basis because factors such as win streak and fighting style could affect a fighter's performance. We were able to predict the outcome of a fighter's next fight without any knowledge of their opponent with 56% accuracy, predict the outcome a fighter's next fight based on their last 5 fights at a 60% accuracy, and use a fighters last 10 fights to predict the outcome of their next 5 fights with a mean accuracy of 57%.**

Keywords—combat sports, data science, deep learning, fight predictor, machine learning, mixed martial arts (MMA), neural networks, ultimate fighting championship (UFC)

I. Introduction

With today's enormous amounts of data and computing power, data science and machine learning is increasingly being used to solve some of the world's most pressing problems. Oil rigs are being equipped with algorithms that autonomously seek out the best drilling parameters to achieve the efficient drilling speeds [1]. Investment banks like JP Morgan employ Natural Language Processing techniques in providing Digital Financial Advisors to their clients [2]. Besides the grittier business applications, machine learning in sports have also became very popular. 47 million Americans will spend an estimated $8.5 million betting on the NCAA basketball championships known as March Madness. Some say you are more likely to win the Powerball jackpot than fill out the perfect March Madness bracket and researchers from around the world have since been trying to employ statistical methods to beat these odds and become the first person in history to create a perfect bracket. From business to sports, the possibilities introduced by Artificial Intelligence and Machine Learning are without bounds, and people are constantly seeking out new applications for these powerful techniques

Our project explores the application of data science and machine learning in the combat sports industry, specifically Mixed Martial Arts (MMA). MMA is a combat sport that allows punching, kicking, grappling, and striking of any kind, using methods and techniques from martial arts and various other combat sports [3]. The goal is to use data science and machine learning techniques to predict the outcome of an MMA fight. This will empower MMA fans and enthusiasts around the world to make more informed decisions when trying to determine the outcome of a fight. This report describes the experimentation procedures and machine learning models that were used in predicting the outcome of MMA fights. It further describes the selection criteria used in determining the optimal models, the results obtained, related work, and the conclusions reached from this analysis.

II. Background

With the UFC and the MMA industry continuing to gain popularity, viewers have begun to place more and more bets. The UFC betting market is clearly one that has been growing over the years and will continue to grow. Datasets containing the statistics of various UFC fights have recently been made available on Kaggle. This study uses the aforementioned dataset along with fight scorecard data to explore how machine learning and data science can be used in predicting the outcome of an MMA fight and an MMA fighter's career trajectory. Being able to predict a fight's outcome and a fighter's career trajectory will enable MMA fans and enthusiasts to make better informed betting decisions.

An MMA fight can end in many ways, but this analysis contains fights that end in only one of two ways: win or loss. Given pre-fight statistics of fighters who partook in numerous UFC fights and fight outcomes, the problem of predicting the outcome of a fight can be reduced to classical machine learning classification task, where fighters are classified as winner or loser based on their statistics up to the current fight. For a more thorough analysis, this classification is attempted in three ways: (1) Predict the outcome of a fighter's next fight without any knowledge or data on the fighter's opponent, (2) Predict the outcome of a fighter's next fight given the fight statistics of the fighter's opponent, (3) Predict the outcome of a fighter's next fight based on the fighter's last N fights. The first classification attempts to predict the outcome of a fighter's next fight solely based on the fighter's own abilities, not taking into consideration the bias that might be introduced by the fight statistics of the

fighter's opponent. The second classification takes these biases into consideration, while the third attempts a prediction on the fighter's current run of form (as opposed to the run of form in the fighter's entire career). This analysis also attempts to predict a fighter's career trajectory, given the statistics of all the fighter's fight to date. A fighter's career trajectory can be represented by the outcome of a number of their future fights, so predicting the outcome of M future fights will be enough to represent a fighter's career trajectory. Insights into a fighter's career trajectory can better inform fans on whether to place bets on particular fighters. For instance, it would be a terrible idea to place a bet on a fighter who is likely to lose 4 of their next 5 fights.

Adopting a data driven approach in Predicting the outcome of a fight is not as straightforward as it might appear. There are certain limitations involved in such an approach. There are certain qualities of a fighter that are hard to measure, and for which no data currently exist. These intangibles include qualities such as the fighters psychological state or mindset, motivational drive, reflexes etc. Also, the course of an MMA fight can be disrupted at any point in the fight. For instance, a fighter on a losing course might land a lucky but hard punch directly to the face of their opponent, and this can instantaneously set him on a winning course. These disruptions are hard to measure, because classifying an attack as lucky is very subjective. Furthermore, it is very hard to measure luck.

This analysis explored models of varying complexity from the basic K-Neighbours Classifier to the more complex Neural Networks. Models such as K-Nearest Neighbours (KNN), Logistic Regression, Support Vector Classifiers (SVC), Naive Bayes, and Random Forests were explored first because these are the models that the authors of this analysis are already familiar with. Different types of neural networks were also explored to capture more sophisticated representations of the data. The performance of these models were judged based on the accuracy and F1 score they yielded on a test dataset. F1 score, in addition to accuracy, was used as a selection criterion because false positives and negatives had to also be accounted for.

III. OBJECTIVES

The goal of this analysis is to predict the outcome of an MMA fight using various approaches. This goal has been broken down into four objectives for a more thorough analysis:

1. Predict the outcome of a fight given the overall statistics of both fighters
2. Predict the outcome of a fighter's next fight without any knowledge or data of their opponent
3. Predict the outcome of a fighter's next fight based on their last N fights
4. Predict a fighter's career trajectory using time-series analysis i.e. predict the results of their next M fights based on their last N fights

IV. DATA COLLECTION

Since we are trying to apply machine learning techniques to predict the outcomes of a fight, we required a statistically significant and diverse dataset that contained data about fights and fighters over a relevant time period. We explored multiple different datasets from various channels and converged on a database on Kaggle [4] that contained all the MMA fights in the UFC from its inception until June 2019. Although this database only contained fights in the UFC and no other organizations, it was deemed to be sufficient for our purposes since the UFC is the major market share holder in this industry. The Kaggle database contained four datasets with different types of data. We only used one of the four datasets because the other datasets had some data cleaning and processing techniques already applied to it and we wanted to apply our own data processing techniques so we decided against using them. The dataset we choose contained 145 columns and around 5000 rows which is not the optimal size for machine learning applications, but it would suffice for our use case. Each row in this dataset contained information about a particular fight; the date, venue, the fighters that participated and the statistics and history of each fighter up until that fight.

There were a few pieces of information that was missing from this dataset, so we scraped and parsed Wikipedia webpages to create two more auxiliary datasets that would provide more context to a fight and a fighter's physiology. One of the new datasets contained more detailed information about each fight in the Kaggle dataset and how the winner of the fight was determined i.e. whether they won by knockout (KO), technical knockout (TKO), decision, etc. The second dataset contained the altitude of every city a UFC event had taken place in and the altitude of the city in which a fighter from the Kaggle dataset trains at. The difference in altitude between where a fighter trains at and where the UFC event takes place is deemed significant for this experiment since there have been several studies conducted that analyze the effects of high-altitude training and how it impacts the performance of elite athletes [5].

V. DATA PREPROCESSING

All of the aforementioned data that was collected had to be cleaned and processed before it could be utilized by the machine learning algorithms. The data that was collected from Wikipedia was parsed using BeautifulSoup and converted into meaningful datasets. The string data columns across all the datasets had to be striped and converted to lowercase. The fighter names across the different datasets were not consistent with each other since some of them contained unicode characters and they all had to be converted to the ASCII standard using the uni-decode method. All of the numerical values were converted from their input data type to float. Date values were also not consistent across the datasets since they were in different time zone and thus, had to be converted to a unix timestamp so they could be compared and merged. Categorical data such as weight classes or how a fighter won was transformed into the categorical data type from strings.

Once the datasets were cleaned, we tried to merge them together and encountered other problems such as the same fighter having different spellings for their name across the three datasets. There were about 1700 unique fighter names in each of the datasets, so it was going to be an impossible task to find the correct spelling manually and merge them together. Therefore, we created an Apache PySpark [6] application that would compute all the unique name combinations between the datasets and score how similar two names were based on the Levenshtein distance [7]. These name combinations, Levenshtein scores, and

dates were then combined together to create a unique key for each fight and fighter. This key was then used to merge the three datasets into the final dataset that was used for all the experiments in this study. This combined dataset contained 5062 rows and 153 columns.

VI. Data preparation

After preprocessing, the resulting data frame contained about 5000 rows. Each row contained data about a fight and fighters involved in the fight. Each objective in this analysis required a unique transformation of the data. This section outlines the objective-unique transformations applied to make the dataset suitable for the purpose of each objective.

A. Objective 1

Objective 1 was concerned with predicting an outcome of a fight given the offensive and defensive statistics of both fighters involved in the fight. Objective 1 was concerned with predicting an outcome of a fight given the offensive and defensive statistics of both fighters involved in the fight. The combined dataset was used for this objective. There were 12 non-numerical columns with string data type which described the fighter names, referee name, date, winner, weight class, fighting stance, detail about how the fight ended, and event location (city and country). The fighter names were removed because it was assumed that names don't play a role in the outcome of the fight. Similarly, referee name, date, and event location columns (but location elevations were kept) were also removed as it was assumed that they did not provide any insight into the result outcome. Lastly, fight ending detail was also dropped because it is impossible to know how the fight would end in advance for the predictions.

The leftover string type columns, namely winner, weight class, and fighting stance were used in this objective. The winner column was used as the label or target for the experiment. Each fight has two corners that are called blue and red. The red fighters are the favorites or champions and the blue fighters are the underdogs or challengers. The winner column consisted of blue or red values and it was converted into a boolean type where 1 meant that red fighter won and 0 meant that the blue fighter won.

B. Objective 2

This objective attempts at predicting the outcome of a fighter's next fight without any knowledge of the fighter's opponent. In other words, this objective attempts to predict the outcome of a fight given the data of only one fighter. As such, each row in the combined dataset had to be broken into two different rows. Each new row contained the fight statistics of a single fighter (as opposed to 2 fighters in the original dataset) and the outcome of the fight.

Certain features in the dataset such as referee, number of rounds, date, end-method, end-round, and attendance were discarded because they were not relevant to this analysis. The rationale is that these features would not be available if the fight is in the future and the opponent is unknown, and thus cannot be inputs to the model. For each fighter, missing values in the age, weight, height, and reach columns were filled with the mean age, weight, height, and reach values in the fighter's weight class. Missing stance values were filled with 'orthodox', because most people tend to be right-handed. Missing elevation values were filled with 0 which is at sea level.

For this objective, three different feature-combinations were tested on the models in an attempt to determine which feature-combination works best. The first feature-combination excludes offensive and defensive features (e.g. average number of punches attempted). Because about 25% of the rows in the dataset were missing offensive or defensive statistics, excluding these features allow for more samples but fewer dimensions. The second feature-combination includes all features, as such rows with missing offensive or defensive statistics were ignored and this reduced the number of samples by about 25%. The third feature-combination includes only the top 50% most correlated features.

C. Objective 3 and Objective 4

These two objectives required the combined dataset to be represented in a unique way since they were peering into a specific slice of a fighter's career and predicting their future without considering their opponent. Since each row in the dataset contained the information about both the fighters in a fight, it had to be split into two rows, where each row contained specific information about each of the fighters and the data that was common between the two, such as date, location, etc. This transformed dataset contained 10124 rows and 95 columns. This dataset was then grouped by each fighter and sorted by date to generate a timeline of their career, from their oldest fight to their most recent. The missing numerical values were then filled with zeroes and categorical data was one hot encoded.

For both Objectives 3 and 4, a feature matrix that would allow for time series analysis was required. Therefore, helper functions were created to slice this dataset into different feature and label matrices to facilitate the deep learning algorithms. Let's consider the scenario where we want to analyze how a fighter's last 5 fights would affect their next 2 fights. All the fighters that had less than 7 fights in their career would be disregarded. Each fighter would then be grouped separately and generate its own feature and label matrices. For example, if a fighter had 8 fights in their career, the details from fights #1 through #5 would become the first row in the 2D feature matrix and the win or loss for fights #6 through #7 would become the first row in the 2D label matrix. The second row in the feature matrix would contain the details from fights #2 through #6 and the second row in the label matrix would contain the win or loss for fights #7 through #8. These feature and label matrices that are unique to each fighter would then be stitched back together to create a combined feature and label matrix, which were used as inputs for the deep learning models. All the numerical values in the features matrix were scaled using StandardScaler, and if there was a value missing, it was replaced with 0.

VII. Data exploration

Most of the data exploration in this analysis were done on the resulting data frames from objectives 1 and 2. Basic dataset statistics such as the number of rows and columns in the data frame were obtained. To get a feel for the amount of valid data samples (rows) available, the number of null values for each feature in the dataset were visualized.

Next, distribution across several categorical features were explored. MMA fights are grouped according to weight classes.

In order to determine if the data set fairly represent each weight class, the distribution of data samples across weight classes were assessed and visualized. MMA fighter's adopt different stances depending on if their left-handed, right-handed, or both. To get a feel of how well our dataset represents various stances, the distribution of data samples across stances were assessed and visualized. The predictions attempted in this analysis are purely data driven and thus can be said to be objective predictions. The fight end-methods were also explored because they provide insights into whether the winner of a fight was decided in an objective or subjective way. For instance, a fight that ended by decision would be considered more subjective than objective, since decisions are based on scores from individual judges.

Finally, correlations between features and fight outcomes were explored. This was done to see which features affect the outcomes of fights the most. Correlations were explored to also prove or disprove common misconceptions among MMA fans and enthusiast. Trends observed across the whole dataset were observed in some more detail, and were examined across individual weight classes to confirm their validity.

VIII. PREDICTION

Several machine learning models were explored for each of the objectives in this project. The reason for using many different models is to contrast and compare the performances of these different models.

A. Objective 1

There were five models used for prediction of this objective. The five models used are the following: Logistic Regression, KNN, SVC, Random Forest, and Dense Neural Networks (DNN). Due to similarities in the process of training and testing, the first four models were grouped and evaluated together, then the best of the four models was compared to the DNN. The following few paragraphs talk about the first four models together and in the end, the procedure used for the DNN is described.

The baseline is the score of four models (Linear Regression, KNN, SVM, and Random Forests) that were trained with the default parameters to test how they performed out of the box. Also, there were no changes made to the dataset, rather the dataset obtained from the preparation stage was used as is. The models were trained on the training set and tested on the test set to evaluate their performance. These performance metrics were used as a baseline for different techniques that were explored to improve upon the results.

The data used for all the different models was obtained by splitting the original dataset into training and testing data sets. The split used was 80/20 where 80% of the original dataset is used for training and 20% for testing. Each model performance was obtained by taking an average over 5 results where each result was computed from a different splitting of the original dataset by using different seed values in train test split function. The evaluation consisted of two metrics: accuracy and F1 score. The details on why this combination was used is described in the Measurements sections later on.

The first technique that was used for model improvement was scaling. Two types of scaling methods were used in this objective: min max, and standard scaling. The scalers were fitted on the training set, and then used to transform both training and testing datasets. The models were retrained on the scaled data, and the accuracy and F1 score were computed to analyze any improvements in the performance of the models. Scaling was used on all models except one, which was Random Forests because scaling generally does not have any effect on Random Forests.

The next technique to improve model performance was parameter tuning. The best parameters were selected using GridSearchCV, which runs the models with different combinations of input parameters and it also does cross validation to ensure proper results. Table 1 below shows the parameters tested for different models, and the best parameters provided by the GridSearchCV. These best parameters were used for each model going into the next and last step of improvement.

The last technique used for model improvement was dimension reduction. The dataset used for this objective had 145 features. Dimension reduction was used to test if the same or better performance can be obtained using only 80% (approx. 120) of the original features. All the techniques were used in a stacked manner, which means that dimension reduction was performed after applying the scaling and using the best parameter obtained from parameter tuning. Two dimension reduction techniques were also attempted: Principal Component Analysis (PCA) and Recursive Feature Elimination (RFE) using a linear regression model.

After exploring the traditional techniques, we implemented a DNN with the goal of outperforming all of the previous techniques. The DNN was trained on the data after it was scaled using Standard Scaler, but before any dimensionality reduction techniques were used. DNNs require tuning of the number of hidden layers, neurons per layer, and the shape of the hidden layers. The number of hidden layers and neurons per layer determined the capacity of the DNN's memory. We attempted different combinations of 0, 1, or 2 hidden layers, 64 or 128 neurons per layer, dropouts of 0, 0.1, or 0.3, and a brick or funnel as the shape. A brick shaped DNN will have the same number of neurons for each hidden layer while the funnel DNN's hidden layers will contain half the number of neurons in every subsequent layer.

TABLE 1. PARAMETER TUNING FOR OBJECTIVE 1

Model	*Parameter Range*	*Best Parameter*
Logistic Regression	Solver: lbfgs, liblinear, sag, saga	liblinear
	Max Iterations: 1000, 5000, 10000	1000
KNN	Weights: uniform, distance	distance
	No. of Neighbors: 2, 3, 5, 7, 9, 12	2
SVC	C: 0.01, 0.1, 1, 10, 100	0.01
	Max Iterations: 1000, 5000, 10000	1000
Random Forests	Max Features: auto, sqrt, log2	auto
	Max Depth: 5, 10, 15	15
	Min Samples Split: 2, 7, 15	2

B. Objective 2

For this objective, the classification models explored were: KNN, Logistic Regression (LR), SVC, Naive Bayes, and Random Forests. As mentioned in the Data Preparation section for Objective 1 above, three different feature-combinations are explored in order to determine which feature-combination results in the best model performance. The feature-combinations explored are: (1) Data without offensive or defensive statistics, (2) Data with offensive and defensive statistics, (3) Top 50% most correlated features. A scaled and unscaled version of each of the feature-combinations was fed into models to see if scaling improved model performance.

For each model used, parameter tuning was done to determine what parameter values yield the highest model performance. Separate Parameter tunings were done for feature-combinations 1 and 2, and the best parameters obtained for these were used on feature-combination 3. Parameter tuning was done with SciKit-Learn's GridSearchCV function. Some experimentation to narrow down parameter ranges was done. Table 2 below shows the parameter ranges used for selection, and the best parameters obtained for both feature-combinations 1 and 2.

C. Objective 3

The classification models explored here were Dense Neural Networks (DNNs), 1-dimensional Convolutional Neural Networks (1D CNNs), and Long Short Term Memory (LSTMs). Each model would take as input the last N fights of a fighter and predict whether or not this fighter will win or lose their next fight. We experimented with using 5, 10, and 15 fights as input into each model. Although they performed the same task, the configurations for each model were very different. Carefully choosing the right hyperparameters was essential for achieving the best results. All neural networks contain the general hyperparameters that can be tuned such as the learning rate, batch size, epochs, and dropouts to avoid overfitting. Besides the general hyperparameters, there are specific hyperparameters to tune for each model due to the different architectures of each such the number of filters for CNNs and the number of memory units for LSTMs.

TABLE 2. PARAMETER TUNING FOR OBJECTIVE 2

Model	*Parameter Range*	*Best Parameter*	
		Comb. 1	*Comb 2.*
KNN	algorithm: auto, ball_tree, kd_tree, brute	ball_tree	auto
	n_neighbors: range(1, 51)	7	7
LR	solver: newton-cg, lbfgs, liblinear	lbfgs	liblinear
	C: linspace(0.001, 20, 40)*	7.1801	0.0010
SVC	C: linspace(0.0000000001, 0.15, 50)	0.0061	0.0490
Naïve Bayes	alpha: linspace(0.0000000001, 21, 50)	21.000	19.7143
Random Forests	max_features : auto, sqrt, log2	auto	auto
	max_depth: linspace(1,30,50)	10.4694	3.3673
	min_samples_split: range(2,30)	15	10

Similar to the DNN trained in Objective 1, we attempted different combinations of 0, 1, or 2 hidden layers, 128 or 256 neurons per layer, and brick or funnel as the shape. The 1D CNNs required tuning of the number of filters in each convolutional layer, kernel sizes, and kernel stride lengths. The number of filters determined how many features to learn for each N fight window and the kernel size and kernel stride length determined how the temporal information would be captured. More specifically, increasing the kernel size to 3 and the kernel stride to 2 causes the convolutional layer to extract information using 3 fights windows and slide the window by 2 fights along the input. For the CNN, we attempted different combinations of 64 or 128 filters, a length of 2 or 3 for the kernel size, and a kernel stride of 1 or 2. LSTMs had one specific hyperparameter to tune which was the number of memory units per LSTM layer. Each memory unit maintained its own state and information about the previous window. We experimented with using 10, 20, 30, 40, and 50 memory units.

D. Objective 4

This objective was a multi-label classification task which analyzed the last N fights of a fighter as input and predicted their next M fights. We experimented with different lengths of inputs and outputs such as 5, 10, 15, and 20 fights as input and predicting up to 5, 6, 7, and 8 fights into the future. To perform multi-label classification, we created a multi-output model which was an ensemble of Neural Networks that could be broken into two separate sections, bottom and top. The bottom was the feature extraction model that would analyze the N fights used as input and extract the relevant information needed for prediction. We were able to use any model such as a DNN, 1D CNN, or LSTM as the bottom feature extractor. We used the best performing model from objective 3 as the bottom which turned out to be the 1D CNN. The top was the final layers that performed the classifications, there were m classifiers at the top layer which were each dense layer that were responsible for predicting the outcome of one fight. For example, if we we're predicting 5 fights into the future, the top layer would have 5 dense layers. One for predicting the result of the next fight into the future, another for predicting the second fight into the future, and so on.

IX. MEASUREMENTS

The simplest measurement of performance for a binary classification is accuracy. As all the models in this experiment tackle binary classification problems, accuracy is used as a primary metric for model performance. Accuracy is simply defined as the total number of correct predictions divided by the total number of predictions. Although accuracy is a simple yet effective way to evaluate model performance, it may not work well with skewed dataset, similar to the one used in this experiment where 63% of wins are towards the red fighter.

Considering the skewed nature of the dataset, F1 score is used alongside accuracy to give a better understanding of the true performance of the models. The F1 score considers both recall and precision and provides a good estimate for skewed binary classification datasets [8]. Before, precision and recall can be defined, confusion matrices need to be introduced.

Confusion matrix further splits the prediction results into 4 classes, namely True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). It can give a good insight into the predictions by showing where the model is getting its right and wrong predictions. The precision is simply TP divided by (TP + FP), and it is used for cases where the number of false positives needs to be limited [8]. Similarly, recall is defined as TP divided by (TP + FN) and it is used where false negatives needs to be controlled [8]. Using both accuracy and F1 score provides a balanced evaluation of the models, therefore these two metrics are used extensively in this study.

For objectives 3 and 4, each model was trained using an 80/20% split for training and validation tests respectively. We would train each configuration of the model on the training test and analyze its accuracy and F1 score on the validation set. For these objectives, we considered a set of hyperparameters better than another if it was able to achieve higher accuracy on the validation set. This was because our models were not able to generalize well on the validation set, so we focused strictly on improving the validation accuracy of our model and not the F1 score. Objective 4 involved predicting the next 5 fights, so we used a mean of the accuracies as the final accuracy of the model. For example, if our model could predict the first fight with 60% accuracy and the second for 50%, our model's final accuracy would be 55%.

X. New techniques

A. *Deep Neural Networks*

As mentioned before, we introduced three deep Neural Network architectures in our experiments. The main advantage we wanted to capitalize on was that Neural Networks have been shown to produce state of the art results without much expert analysis or fine tuning [9]. DNNs were used when we did not need to maintain any of the temporal information of the data. In contrast, 1D CNNs maintained the temporal dependencies by employing a sliding window across the input which has been shown in previous literature to produce excellent results in Time-Series analysis [10]. We also used LSTMs because they also employ a sliding window but maintain a state of the last window they analyzed to use as input to the next window [11].

To develop these Neural Networks, we used Keras with a TensorFlow backend. The main reason for using Keras was that it allowed for rapid prototyping with a simple sequential or functional API. The sequential API was used to develop the DNNs, CNNs, and LSTMs while the multi output model required a fancier configuration which we were able to create using the functional API. Keras also offered enough flexibility by providing common predefined functions for optimizers, loss functions, and activation functions.

B. *Talos for Hyperparameter Optimization*

To efficiently test all possible combinations of hyperparameters for each model, we used Talos to automate the training and evaluation of our models. To use this library, we had to create a dictionary of the hyper parameters we wanted to try for each model (see Figure 1). This dictionary was used as input into the Talos Scan function to generate all permutations of DNNs and train them. When it was finished executing, it returned us a pandas data frame of the results for each configuration. The results data frame contained all relevant

```
dnn_params = {'lr': [0.01, 0.05, 1],
              'first_neuron': [64, 128, 256],
              'activation': ['relu'],
              'hidden_layers': [1, 2, 3],
              'batch_size': [32, 64, 128],
              'epochs': [10, 15, 25],
              'dropout': [0.1, 0.2, 0.5],
              'shapes': ['brick', 'funnel']}
```

Figure 1: Talos Hyperparameters used as input into the Dense Neural Network.

metrics including the training accuracy, validation accuracy, hyper parameters used, precision, recall, and F1 score.

XI. Exploratory Analysis Results

The dataset used for this objective consisted of 5,062 rows and 153 columns, and this translates to 774,486 data points. Out of these total number of data points, 96,205 were null values, which is approximately 12.4% of the total data points. It was found that only 75% of the dataset had complete data. Also, offensive and defensive statistics were missing for 49 out of 153 columns. This means about 1,240 rows out of 5,062 total (25%) were missing 49 values or more.

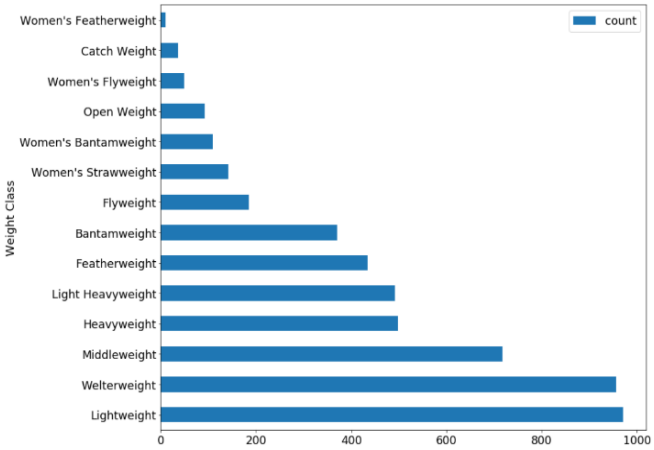

Figure 2: Weight class distributions

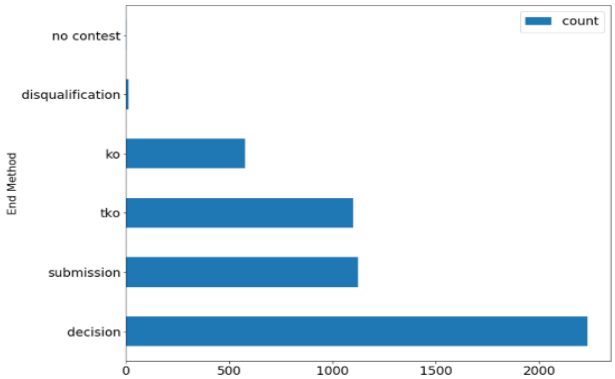

Figure 3: End method distributions

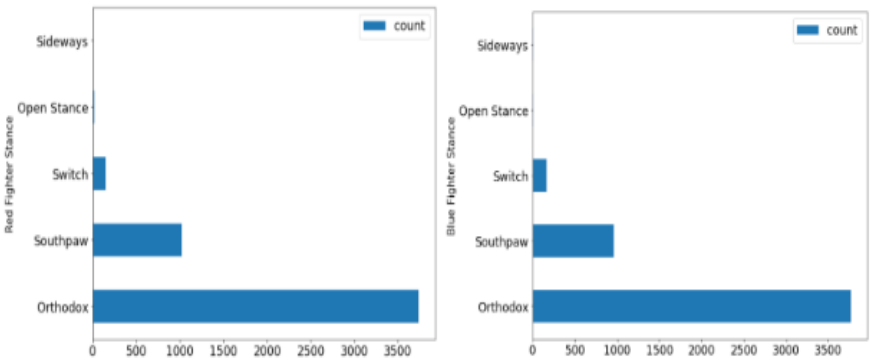

Figure 4: Red and blue fighter distributions

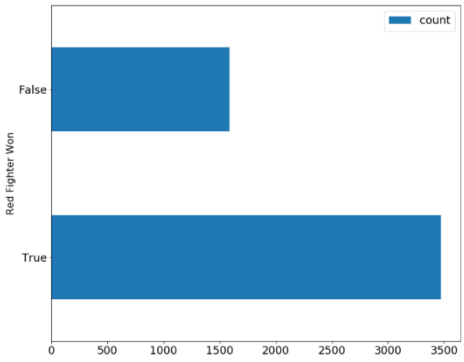

Figure 5: Winner distributions between red and blue fighters

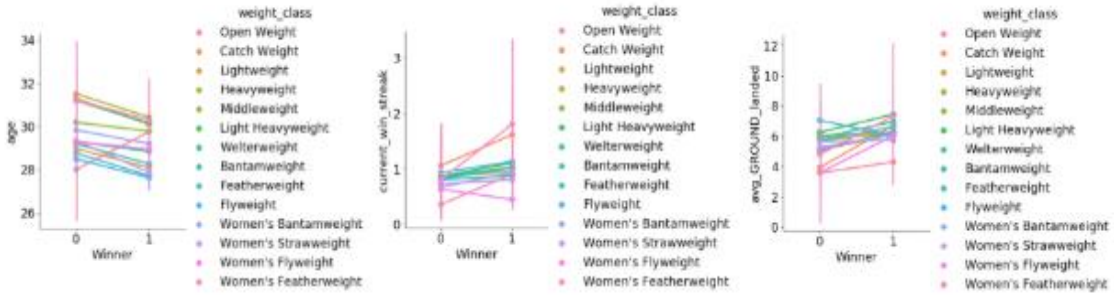

Figure 6: Age, win streak, and ground attacks trends across weight classes

The data distribution across weight classes were satisfactory. Simply looking at figure 2, it is easy to see that the dataset contains a fair number of samples from all male and female weight classes. With regards to fight end-methods, figure 3 shows that most of the fights in this dataset ended by decision. When a fight ends by decision, each of the judges assign the fighters a score based on how well they think the fighters have fought, and the fighter with the highest score wins. A decision end-method can vary by judge and can also very subjective. A fighter who wins by decision is not necessarily the better fighter. This might introduce some inconsistencies in our prediction, as our analysis assumes the better fighter wins. With regards to stance, figure 4 shows that the dataset is not really representative of fighters with sideways and open stances. This is not too much of an issue because very few fighters adopt those stances.

The features that correlated most to fight outcomes were the features related to the fighter's age (9% correlation), ground attacks (7% correlation) and streaks (5% correlation). Figure 6 looks deeper into fighter age, ground attack, and win streaks. From figure 6, we see that younger fighters, fighters with more successful ground attacks, and fighters with higher win streaks tend to be winners. Contrary to popular opinion, we found that a fighter's height and reach does not really contribute to the outcome of a fight. Figure 7 below, shows that the distribution of height and reach across weight classes is the same for both winners and losers, and in contrast it shows winners tend to be younger.

Lastly, the exploration analyzed the winner column to check for a skewed dataset. A red fighter is regarded as the favorite or champion and blue fighter is regarded as the underdog or the challenger, therefore, the data was expected to be skewed

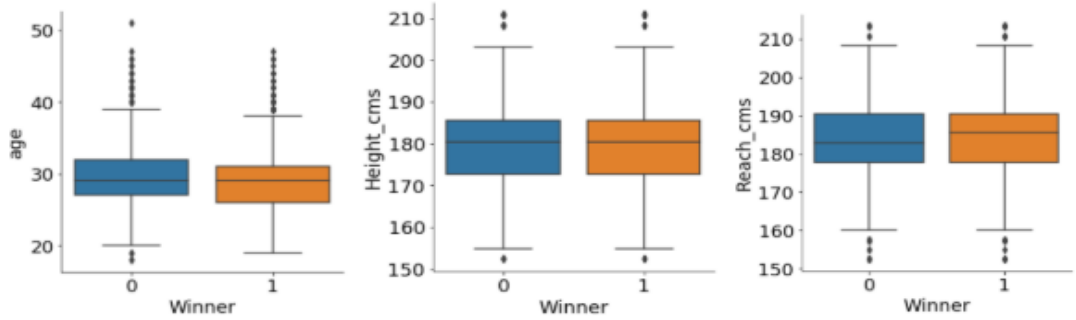

Figure 7: Winner's age, height, and reach distributions

towards having more wins for the favorite fighter and we found this to be true. Figure 5 shows that the dataset was skewed towards the red fighter who won 63% of the time.

XII. Predictive Analysis Results

A. *Objective 1*

The accuracy and F1 score of the four models (Logistic Regression, SVC, KNN, and Random Forests) are compared in figures 8 and 9. Figure 8 shows the F1 score improvement of the four models through the different improvement techniques described in the Prediction section of Objective 1. In both figures 8 and 9, the scaling refers to standard scaling and dimension reduction refers to RFE with linear regression. These were better options in their groups, so they have been chosen here for this comparison. Logistic Regression, SVC, and KNN have an identical outcome with respect to the techniques. The most important technique for these three models is scaling, which seems to have improved the F1 score for these three models. However, the same could not be said for parameter tuning and dimension reduction because these three models do not improve using these techniques. Even though the F1 score for Logistic Regression, SVC, and KNN don't improve or worsen with dimension reduction, it still should be considered a success because the models are able to perform at the same level using fewer features. The same could not be said for Random Forests, because its F1 score got worse after applying parameter tuning and dimension reduction. Logistic Regression and KNN are tied for the best model for F1 score. In figure 9, the accuracy of the four models (Logistic Regression, SVC, KNN, and Random Forest) is compared. The accuracy is stable for Logistic Regression and Random Forest over different techniques. However, the accuracies of SVC and KNN models mostly improved from parameter tuning and scaling respectively. The best model for accuracy was Logistic Regression at around 71% accuracy.

Considering both accuracy and F1 score, Logistic Regression seems to be the best of the four models (Logistic Regression, SVC, KNN, and Random Forest). The F1 score for Logistic Regression is 60% and accuracy is 71%. We took the Logistic Regressor and compared it to our best DNN. The best configuration of hyperparameters for the DNN was able to

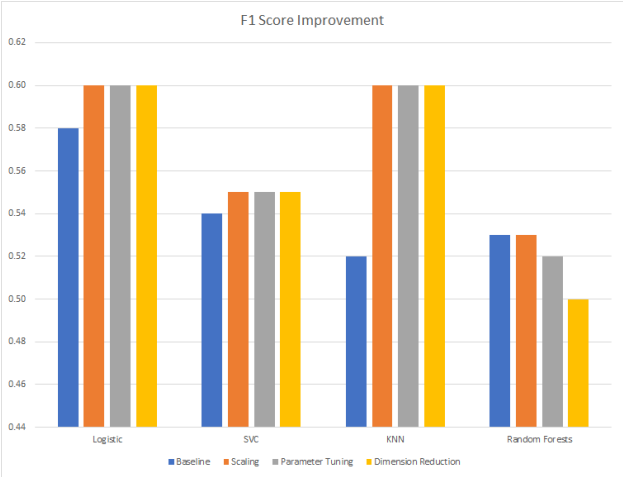

Figure 8: F1 Score improvements of different models using different techniques

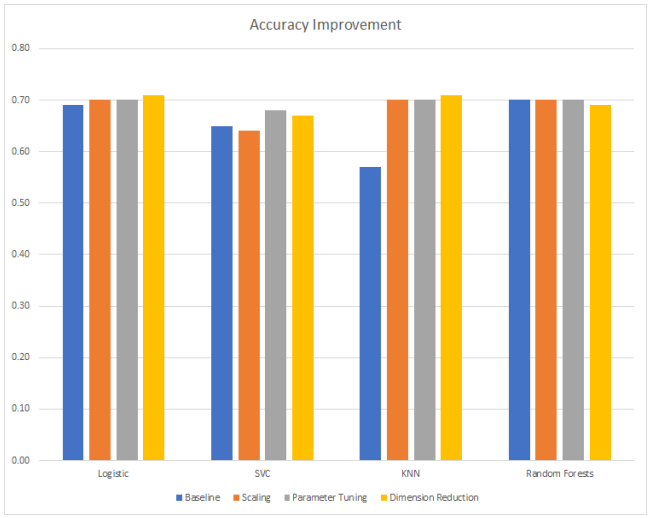

Figure 9: Accuracy improvement of different models using different techniques

achieve 72% accuracy and 80% F1 score. We found that adding too many neurons would cause the DNN to learn all possible mappings of the input and would often overfit on the training data causing a decrease in the validation data. For example, a DNN with 2 hidden layers of 128 neurons trained for 25 epochs was able to achieve 99% accuracy on the training data, but only 67% on the validation set. The best performing model had only one hidden layer of 128 neurons with a funnel shaped configuration trained for 10 epochs. Table X shows the top 3 DNNs based on test accuracy.

B. *Objective 2*

It was found that scaling the dataset tended to reduce accuracy and F1 scores for all models by about 1.5%, hence the input data frame used in this objective was left unscaled. The best feature-combination includes all the available features in the dataset (feature-combination 2). Excluding offensive or defensive statistics (feature combination 1) reduced model accuracy and F1 score by about 1%, while using only the top 50% most correlated features reduced model accuracy by about 1.5%.

SVC was excluded from the model selection because it produced very inconsistent results, sometimes achieving a 100% accuracy and other times achieving 0% accuracy across 20 folds

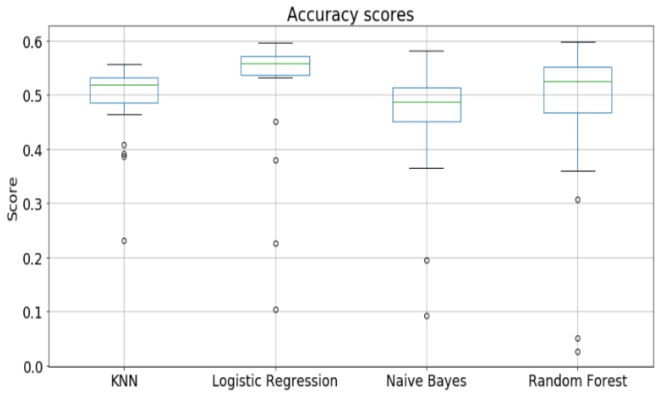

Figure 10: Accuracy score distribution across models

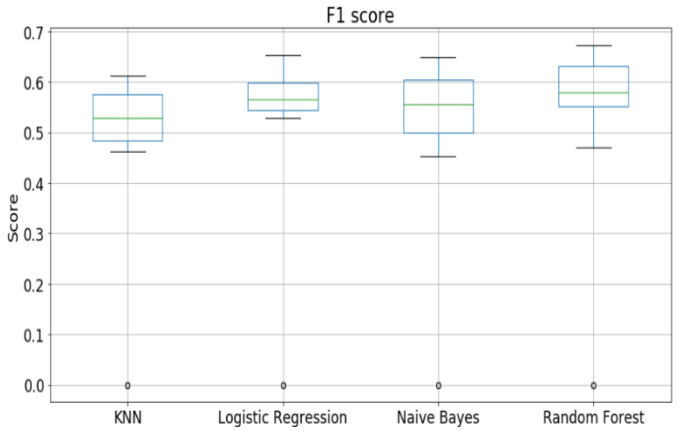

Figure 11: F1 score distribution across models

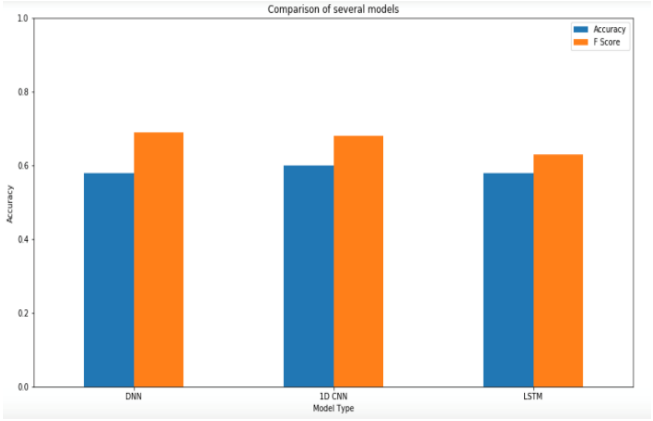

Figure 12: 1D CNN, and LSTM with respect to their accuracies and F1 scores.

of the dataset. Figure 10 and 11 above show the box plots of accuracy and F1 scores for the other models considered across 20 folds of the dataset. The Logistic Regression model performed the best of all the model used in terms of accuracy and is considered better than the Random Forest classification because the F1 scores it produced across iterations were more consistent. The median values obtained for train accuracy, test accuracy, and F1 score for the Logistic Regression model over 20 folds are 58%, 56%, and 56% respectively.

C. *Objective 3*

For objective 3, we experimented with varying ranges of N fights as input into our models. We noticed that analyzing more than 5 previous fights such as 10 or 15 caused our models to underfit on the training set. Furthermore, using more than 15 previous fights was unfeasible because many UFC fighters have less than 15 fights. After this initial experiment, we decided to go with 5 fights as input into our model. As shown in figure 12, every model performed approximately the same in accuracy (58-60%). The following results are the best performing models.

1) Dense Neural Networks: The best configuration of hyperparameters for the DNN was able to achieve 58% on the test set. Similar to objective 1's DNN, adding too many neurons would often overfit on the training data causing terrible results in our validation data. The best performing model had only one hidden layer of 128 neurons with a funnel shaped configuration. Learning rate and batch size was also essential, and we would often see a drastic shift in the model's accuracy depending on these values. The optimal values were 1 for learning rate and 64 for the batch size. Furthermore, we noticed that overfitting was an issue with our small dataset and the top 5 models handled this by applying a dropout of 0.1.

2) 1-Dimensional Neural Networks: The best configuration of hyperparameters for the CNN was able to achieve 60% on the test set. Similar results to the DNN where a batch size of 64 and dropout of 0.1 was optimal. More interestingly, the best configurations used a kernel size of 2 and a stride of 1 which means that the model was analyzing two fights at a time and analyzed every combination of sequential fights.

3) Long Short Term Memory: Changing the number of memory units had little effect on the accuracy with each configuration hovering around 57% accuracy. The best performing model had 50 memory units and was able to achieve 58% accuracy on the test set.

D. *Objective 4*

Similar to objective 3, we had to search for the optimal amount of previous fights to analyze. Since we were predicting career trajectories, we decided to use a longer time frame of a fighter's past. Most fighters' fight 2-3 times a year, therefore using 10 previous fights would be approximately encapsulate the previous 5 years of a fighter's career and predicting the next 5 fights would approximately look 2 years into the future for that fighter. The multi output model was able to predict the accuracy for the 1st fight into the future with 63%, 2nd fight with 55%,

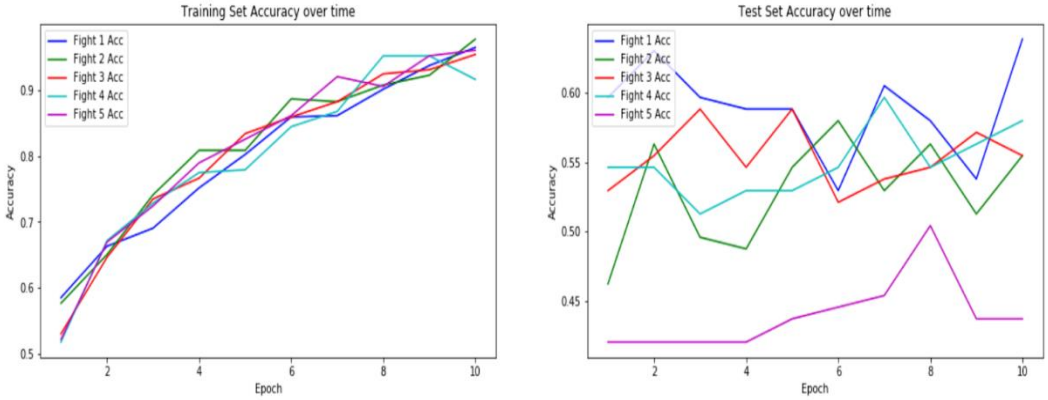

Figure 13: Training and Test set accuracy across epochs for the multi-output model

3rd fight with 55%, 4th fight with 57%, and 5th fight with 43%. The mean validation accuracy was 54.6%. These results seem reasonable because the further the model looked into the future, the less reliable its prediction became. As shown in figure 13, when looking into the performance of the models during training, we could see that our models were consistently increasing in accuracy on the training set with every epoch, but the validation accuracy was fluctuating. This could be due to our small dataset because our models were strong enough to learn mappings in the training dataset but was never able to learn any mapping in our validation set.

XIII. Discussion on results

As already mentioned, Objectives 1, 2, and 3 attempts to predict the outcome of a fight in different ways. Without doubt, predicting the outcome of a fight based on the statistics of only one fighter (Objective 2) is the most unreliable method of the bunch with an accuracy and F1 score of 56%. This comes as no surprise seeing that the likelihood of a fighter winning a fight heavily depends on the quality of the fighter's opponent. A fighter is more likely to win a fight against an inferior opponent and is more likely to lose a fight against a superior opponent. Predicting the outcome of a fighter's next fight based on the fighter's last N fights (Objective 3) performed slightly better with a mean accuracy of 60%. Again, this is expected given that the last N fights of the fighter is considered when attempting a prediction (as opposed to Objective 2 which only takes current fight statistics into account). We believe the last N fights helps establish a trend of the fighter's likelihood to win which the Neural Network benefits from. Predicting the outcome of a fighter's next fight with the knowledge of the fighter's opponent (Objective 1) emerges as the best fight prediction method with an accuracy of 72% and an F1 score of 80%. Knowledge of the fighter's opponent provides a lot of context for prediction. The availability of the opponent's data provides for very easy benchmarking of fighters, and this makes it a lot easier for the Neural Network to learn the patterns which makes a fighter emerge as the winner of a fight.

Predicting the outcome of a fighter's next M fights based on the fighter's last N fights (Objective 4) is a slightly harder problem. This is because the next M opponents of a fighter are very rarely known which means the analysis of the N previous fights had to be done without the fighter's opponent statistics. Furthermore, we already established that predicting the outcome of a fight based on the data of only one fighter is not the optimal way of predicting fight outcome. The multi output model was able to predict the accuracy for the 1st fight into the future with 63%, 2nd fight with 55%, 3rd fight with 55%, 4th fight with 57%, and 5th fight with 43%. Overall, accuracy falls as we attempt to predict fights farther into the future. This is expected as fights farther into the future are more out of context than those nearer to the present.

XIV. Related Work

Previous publications in MMA include the use of data-driven analysis to understand which styles win fights but these papers do not perform any predictions [12]. Little et al used traditional statistical models to analyze facial cues and predict the winner of MMA fights [13] but there has never been any publicly published work of machine learning in MMA. There have only been blogs, Kaggle posts, and class projects of others attempting to use machine learning technique such as SVMs and DNNs to predict the fight outcome [14, 15, 16]. The work done by J. Chan and J. An [17] closely relates to the work done under Objective 1 of this experiment. Both Jason's web application and Objective 1 try to predict the winner of a fight using some basic statistics about the two fighters. Both use a variety of models for comparison and prediction, where some models like Logistic Regression, Random Forest, and Neural Network are identical in both cases. Another similarity is that both works use the data whose labels are skewed towards the favorite fighter winning around 63% of the time. Lastly, both Jason's work and Objective 1 try out some type of dimension reduction and hyperparameter tuning to improve the performance.

However, that's where the similarities end between Jason's work and Objective 1. The main difference between them is the size of the feature set. Jason's work uses only 10 features for each fighter whereas Objective 1 uses an upward of 150 features. The dataset used by both works is quite different. Another major difference between the Jason's work and this experiment is that the Jason only looks at this problem from one angle, which is to predict the winner of a fight. However, our experiments explore it from four different angles (the four objectives). Lastly, they only evaluate the performance of the models using accuracy whereas Objective 1 uses a combination of both accuracy and F1 score.

Most of the work that motivated our choice of techniques came from papers that were predicting the outcome of sports events in general. Pablo Bosch compared classical machine learning algorithms against deep learning techniques in predicting the outcomes of American Football games but was unable to obtain a better accuracy than SVM, Random Forests, and Logistic Regression models [18]. Petterson et al showed more promising results when predicting the winner of a soccer match [19]. Similar to our approach, they began with a baseline model that predicted the outcome between two teams with no historical input. Afterwards, they began feeding the network with more historical information and were able to increase their prediction accuracy from 33.35% to 88.6% when Time-Series was infused into the dataset. This indicated the potential for Deep Learning in sports prediction.

XV. Conclusion

With the recent popularity of machine learning in sports, our project explores the application of data science and machine learning in Mixed Martial Arts. The Kaggle dataset used for our experiments contained the MMA fights in the UFC from its inception until June 2019. More detailed information such as the outcome of the fight (KO, Decision, or Submission) and fighter training altitude was injected into the dataset to enrich its features. Once this data was collected, various preprocessing and feature engineering techniques were empirically tested to obtain the optimal representation of the data. Each of our objectives required different transformations of the data and different machine learning algorithms. DNNs were able to predict the outcome of a fight between two opponents with a 75% accuracy. A Logistic Regression model was able to predict the outcome of a fighter's next fight without any knowledge of their opponent with 56% accuracy. Using Time-Series analysis, 1D CNNs we're able to predict the outcome a fighter's next fight based on their last 5 fights at a 60% accuracy. Lastly, a multi-output model

with a 1D CNN as a base and 5 Dense Layers as the top was able to use a fighters last 10 fights to predict the outcome of their next 5 fights with a mean of 57% accuracy.

XVI. Future work

There were a lot more experiments and analysis we wanted to conduct on these datasets, but we were not able to accomplish them for the purposes of this paper due to the time constraints. In all the different feature sets that were created for the different models, the fighter name columns were not used in any of them. We wanted to re-run all the best models above but with the fighter names added to the feature set and analyze how our predictions changed. This would be a demonstration in whether there are innate qualities to a fighter that cannot be captured by quantitative measures. There are also opportunities to improve the models by testing a more diverse set of configurations. The bias in the feature sets could also be improved if we allocated more time towards feature engineering and feature reduction techniques.

For some of the other experiments we considered, other datasets would be required, such as fight data from organizations other than the UFC, a detailed breakdown of a fighter's health and physical condition, a fighter's social media presence and, the revenue and viewership numbers for an event. Using these new fights and the fighters' health condition datasets we would be able to diversify our training and design new machine learning models that can incorporate a fighter's health and physical conditions into the prediction of a fight. This would also allow us to generate a list of key performance indicators that constitute a winner or champion and MMA fighters would be able to train differently using these metrics. By analyzing a fighter's social media presence and the historical revenue and viewership numbers they have generated, we could train new models to predict the expected revenue a theoretical fight would generate and whether the fans would be excited for such a fight. This would allow MMA promotional companies to better to make better financial decisions and understand their audience more.

References

[1] O. Bello, C. Teodoriu, T. Yaqoob, J. Oppelt, J. Holzmann and A. Obinwanne, "Application of Artificial Intelligence Techniques in Drilling System Design and Operations: A State of the Art Review and Future Research Pathways", *https://www.researchgate.net/publication/304996042_Application_of_Artificial_Intelligence_Techniques_in_Drilling_System_Design_and_Operations_A_State_of_the_Art_Review_and_Future_Research_Pathways*, 2016.

[2] C. Hudson, "Ten Applications of AI to Fintech", *https://towardsdatascience.com/ten-applications-of-ai-to-fintech-22d626c2fdac*, 2018.

[3] Wikipedia, https://en.wikipedia.org/wiki/Mixed_martial_arts, 2019.

[4] R. Warrier, "UFC-Fight historical data from 1993 to 2019", *https://www.kaggle.com/rajeevw/ufcdata*, 2019.

[5] John P. Porcari, Lauren Probst, Karlei Forrester, Scott Doberstein, Carl Foster, Maria L. Cress and Katharina Schmidt, "Effect of Wearing the Elevation Training Mask on Aerobic Capacity, Lung Function, and Hematological Variables", in *Journal of sports science & medicine vol. 15,2 379-86*, 2016.

[6] Spark, *https://spark.apache.org*, 2019.

[7] N. Babar, "The Levenshtein Distance Algorithm", *https://dzone.com/articles/the-levenshtein-algorithm-1*, 2018.

[8] A. Müller and S. Guido, "Introduction to machine learning with Python: A Guide for Data Scientists (1st ed.)", 2016.

[9] N. O. Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. Velasco-Hernandez, L. Krpalkova, D. Riordan and J. Walsh, "Deep Learning vs. Traditional Computer Vision" in *Advances in Computer Vision Proceedings of the 2019 Computer Vision Conference (CVC). Springer Nature Switzerland AG, pp. 128-144*, 2019.

[10] R. Assaf and A. Schumann, "Explainable deep neural networks for multivariate time series predictions" in *IJCAI International Joint Conference on Artificial Intelligence*, 2019.

[11] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks", *arXiv preprint arXiv:1909.09586*, 2019.

[12] S. R. Hackett and J. D. Storey, "Mixed Membership Martial Arts: Data-Driven Analysis of Winning Martial Arts Styles" in *http://www.sloansportsconference.com/wp-content/uploads/2017/02/1575.pdf*, 2017.

[13] A. C. Little, V. Třebický, J. Havlíček, S. C. Roberts and K. Kleisner, "Human perception of fighting ability: Facial cues predict winners and losers in mixed martial arts fights" in *Behavioral Ecology by Oxford University Press pages 1470-1475*, 2015.

[14] Y. Tian, "Predict UFC Fights with Deep Learning", *https://medium.com/@yuan_tian/predict-ufc-fights-with-deep-learning-e285652b4a6e*, 2018.

[15] A. Saabas, "Who are the best MMA fighters of all time. A Bayesian study", *https://blog.datadive.net/who-are-the-best-mma-fighters-of-all-time-a-bayesian-study/*, 2015.

[16] K. Aggarwal, "UFC Predictor and Notes", *https://www.kaggle.com/calmdownkarm/ufc-predictor-and-notes*, 2017.

[17] J. Chan and J. An, "UFC MMA Predictor Workflow", *https://github.com/jasonchanhku/UFC-MMA-Predictor/blob/master/UFC%20MMA%20Predictor%20Workflow.ipynb*, 2019.

[18] P. Bosch, "Predicting the winner of NFL-games using Machine and Deep Learning" in *https://beta.vu.nl/nl/Images/werkstuk-bosch_tcm235-888637.pdf*, 2018.

[19] D. Pettersson and R. Nyquist, "Football Match Prediction using Deep Learning" in *https://pdfs.semanticscholar.org/e556/af01e86c3414042aa69831ea5fb398e66f94.pdf*, 2017.

CONTRIBUTIONS

A. Abdulkareem Dolapo

Abdulkareem did some of the basic dataset exploratory analysis, all of the exploratory analysis concerned with feature correlations, and all of the deeper analysis into various features on a per weight class basis to see how they influence fight outcome. Abdulkareem was also responsible for all analysis done in Objective 2 of this project.

B. Anil Sood

Anil did the web scraping for the elevation of the fight location, fighter's hometowns, and elevations of fighter's hometowns. He also joined the fight location elevation data to the original dataset. Then he worked on some basic exploratory analysis on null values and categorical columns. Lastly, he was also responsible for all analysis, except DNN, done in Objective 1 of this project.

C. Satyaki Ghosh

Satyaki Ghosh did the web scraping for the fight scorecard details from Wikipedia. He also joined the cleaned and joined the different datasets together by creating the pyspark application with the levenshtein distance computation. He was responsible for creating the different datasets to be used by the neural networks in Objective 1, 3 and 4 and worked on the DNN for Objective 1 and the different neural networks in Objective 3. Satyaki also performed hyperparameter tuning for the DNN used in objective 1.

D. James Peralta

James Peralta was responsible for researching and implementing the Neural Networks done in this project. He created DNN's, RNN's, and multiple output models to run predictions on objectives 1, 3, and 4. James also performed hyper parameter tuning for the models in objective 3 and objective 4.

Signed (Digitally) on the 12th of December 2019:
Abdulkareem Dolapo AD
Anil Sood AS
Satyaki Ghosh SG
James Peralta JP