ISSN 1330-3651 (Print), ISSN 1848-6339 (Online) https://doi.org/10.17559/TV-20240307001386
Preliminary communication

Intelligent Prediction of the Sport Game Outcome Using a Hybrid Machine Learning Model

Kaiwen CUI *, Xuanyi LI, Shuo YANG

Abstract: The National Collegiate Athletic Association (NCAA) serves as the platform for showcasing the skills of talented basketball players from various colleges. With the historical set provided by NCAA this study proposes a hybrid model which is combining the gradient boosting decision tree (GBDT), Tabnet and support vector machine (SVM) for 2023 NCAA basketball game outcome. For each possible matchup between two top college teams, the model can predict the probability of the win rate and the winner team. The fusion model combines the strengths of tree-based model, linear models like SVM and Tabnet to enhance prediction performance, robustness, and interpretability. The data exploration and preparation part shows the important features like the win Ratio of different teams and the feature engineering for the further model training.The experiment part shows the data distribution and feature engineering and performance for each model. The hybrid model beats the separated model with a better brier score of 0.176, which shows the superiority of the hybrid model.

Keywords: basketball game; GBDT; hybrid model; NCAA; SVM; Tabnet

1 INTRODUCTION

The NCAA (National Collegiate Athletic Association) is a non-profit entity responsible for governing student athletics within a network of approximately 1100 educational institutions [1]. The NCAA Basketball Tournament, commonly referred to as the NCAA Tournament or March Madness, is a highly anticipated annual event in American college basketball. Founded in 1939, the tournament showcases the intense competition among 68 top-tier collegiate teams from across the country. The tournament adheres to a single-elimination structure, in which teams engage in a series of matches to determine the national champion. The selection of participating teams is achieved through a combination of automatic bids, awarded to conference champions, and at-large bids, determined by a selection committee that considers factors such as the teams' regular season performance, strength of schedule, and other relevant criteria. The NCAA games can be divided into Men's and Women's basketball tournaments games. In this paper, we utilized historical data from NCAA basketball games, including both men's and women's exhibition matches and tournaments, which can be downloaded from the Kaggle website [2].

As machine learning continues to advance, numerous artificial intelligence (AI) technologies have found their application in many areas [3-6]. These AI-driven solutions have proven instrumental in aiding athletes to enhance their performance and predicting the outcomes of sporting events. By leveraging the power of machine learning, athletes can receive invaluable insights to improve their skills and maximize their scoring potential. In paper [7], the contributions of it lie in its novel application of ML techniques to address the crucial problem of player position classification in football. By utilizing a stacked ensemble model and carefully selecting relevant features, the study provides valuable insights and a reliable framework for coaches and analysts to effectively identify player positions. These findings have significant implications for player recruitment, team formation, and game strategy optimization in the sport of football. The machine learning can also help athletes for their sports injury prediction [8, 9]. C Huang [8] presents a novel approach using Artificial Neural Network (ANN) to develop and implement an early-warning system for injury prediction based on exercise load and performance data. The aim is to construct a hierarchical machine learning prediction system capable of accurately detecting player injuries at an early stage. By identifying and addressing the specific needs of athletes prone to injury, the system can help prevent potential harm and reduce physical strain. The paper [9] developed a hierarchical machine learning model for injury prediction based on athletic load data. The system effectively identifies players at risk of injury, enabling early interventions to mitigate potential harm. This innovative approach has the potential to revolutionize injury prevention strategies in the sports industry, optimizing athlete performance and well-being. The articles [10, 11] show the possibility of the application in predicting the outcome of sport competitions. The article [10] introduces a unique framework for sports prediction that combines Bayesian inference, rule-based reasoning, and an in-game time-series approach. Its contributions lie in addressing the challenge of limited data, considering multiple relevant factors, and capturing the dynamics of sports matches. The implementation and evaluation of the FRES system validate the effectiveness and stability of the proposed framework, paving the way for improved sports prediction methodologies. The contributions of the research [11] lie in the development of an intelligent machine learning framework specifically tailored for NBA game outcome prediction. By uncovering the influential features through a comparative analysis of different machine learning methods, the study sheds light on the factors that significantly impact game results. This information can be invaluable for sports analysts, betting enthusiasts, and stakeholders seeking to improve prediction accuracy and make more informed decisions based on historical data. Another research [12] also presents a novel intelligent machine learning framework for predicting the outcomes of National Basketball Association (NBA) games. Various machine learning methods, including Naïve Bayes, artificial neural network, and Decision Tree, are employed to derive predictive models using different learning schemes.

However, most of the aforementioned works primarily employed traditional machine learning models like Naïve Bayes, support vector machine [13] and gradient boosting

decision tree (GBDT) [14] rather than deep learning models. Compared to deep learning, machine learning method is often easier to interpret and understand the results of machine learning models, making them more transparent. Additionally, machine learning models require less computational resources and are more computationally efficient, making them suitable for deployment on resource-constrained devices. However, machine learning has limitations. It heavily relies on feature engineering [15], which can be time-consuming and requires domain expertise. Machine learning models may struggle to capture complex patterns and relationships in data due to their shallower architectures, leading to potentially lower performance in tasks requiring sophisticated representations. In contrast, deep learning [16-19] excels in capturing complex patterns and automatically learning hierarchical representations from raw data. It achieves state-of-the-art performance in various domains, particularly in computer vision [20, 21] and natural language processing. Deep learning models often require large amounts of labelled data for training and significant computational resources, and they are often considered as black boxes lacking interpretability. With the rapid growth of computing resources, especially the computational power of GPUs, and the availability of deep learning training and inference frameworks like PyTorch [22] and Tensorflow [23], the previous bottlenecks of deep learning have been resolved. In paper [20], authors employ neural architecture search to create a novel baseline network and then scale it up, resulting in a collection of models known as EfficientNets. These models surpass previous ConvNets in terms of both accuracy and efficiency. Notably, the EfficientNet-B7 achieves an impressive top-1 accuracy of 84.4% and top-5 accuracy of 97.1% on ImageNet. Due to its accuracy and efficiency, EfficientNet becomes the backbone model for computer vision tasks for a period of time. The latest breakthrough in computer vision domain is SEEM [21] a promptable and interactive model for segmenting everything everywhere all at once in an image. This research introduces an innovative decoding mechanism that allows for versatile prompting in all kinds of segmentation tasks and a universal segmentation interface that mimics the behaviour of large language models (LLMs). In NLP area, deep learning models like Bert [24] and GPT [25] outperform the traditional models in all NLP tasks like text classification, text summary, question and answers. BERT is a bidirectional language representation encoder model built with transformer. BERT is specifically designed to pretrain deep bidirectional representations from unlabeled text, taking into account both left and right context at all layers. This unique approach enables the pre-trained BERT model to be fine-tuned with just one additional output layer, resulting in state-of-the-art models for various tasks, including question answering and language inference, without requiring significant task-specific architecture modifications. Unlike BERT that focuses on capturing bidirectional contextual information, GPT excels at generating coherent and contextually appropriate text. GPT's subsequent models, such as ChatGPT [26], are large language models trained using unsupervised pre-training and RLHF (Reinforcement Learning from Human Feedback) training. Their emergence has sparked a new wave of generative AI, which will usher in the realization of more artificial intelligence products. Although deep learning achieves success in computer vision and natural language processing areas, it still cannot process the tabular data well before the outcome of Tabnet [27]. Tabnet is a well-designed high-performance and interpretable canonical deep tabular data learning model and it outperforms many other models like Xgboost [8], Lightgbm [9] and Gradient Boosting Decision Tree. Tabnet has many applications in many machine learning areas like stock return prediction [28, 29], depression prediction [30] and rain forecast [31]. To summarize,
despite the fact that there are many new methods in the field of deep learning, for the problem of NCAA basketball prediction, tree-based methods such as Xgboost, LightGBM, and GBDT, as well as models like Tabnet that are specifically designed for tabular data, have shown better performance.

Our model is a fusion of traditional machine learning model like gradient boosting decision tree and the TabNet. It has a better performance than singular model method in terms of accuracy and Brier score. The method we propose can be used to predict the outcomes of sporting events, such as NCAA basketball games, rugby, soccer, and other types of matches. This can enable teams to improve their chances of winning through better data analysis. The main contribution for our works is as follows:

(1) Data analysis for NCCA tourney and regular competitions and corresponding feature engineering for extracting the useful features. These features are carefully designed for training models.

(2) Build a hybrid model with customized Tabnet, GBDT and support vector machine.

(3) Demonstrate the superiority of the model in experiment part and give the top features using feature importance score.

To better illustrate our work, the remainder of this paper is organized as follows:

(1) Exploratory data analysis: data introduction and feature distribution.

(2) Data preparation: data preprocessing for model training

(3) Algorithms: several algorithms are illustrated including hybrid model;

(4) Experiments and Evaluation: Models are evaluated in the experiments and assessed according to the predefined goals, criteria, and metrics.

2 EXPLORATORY DATA ANALYSIS

For this work, The NCCA basketball game data [2] can be divided into three kinds, NCAA Tourney Seeds data, Regular Season Compact Results data and NCAA Tourney Compact Results data. For TourneySeedsdata, it contains Season, Seed and TeamID columns. Season means the year that the tournament was played in. Seed is a 3/4-character identifier and the first character of it is either W, X, Y, or Z (identifying the region the team was in) and the next two digits (either 01, 02, ..., 15, or 16) tell you the seed within the region. The last column is TeamID which means id number of a team.

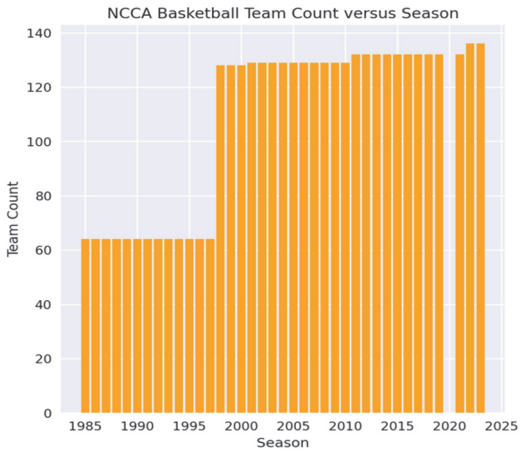

Figure 1 NCCA team count versus season

Fig. 1 shows the distribution of NCCA Team Count versus Season. The figure shows that the NCAA basketball game is held from 1985 to 2023 and the team count is double after 1998. That is because NCAA season for men started from 1985, while the women season started from 1998. Regular Season Compact Results data and Tourney Compact Results data are from different competitions but have the same data format. They both have the following columns: Season, DayNum, WTeamID, WScore, LTeamID, LScore, WLoc and NumOT. Season has just been explained in the above context and the meaning of other columns is going to be introduced. DayNum denotes the range of this integer is always between 0 and 132, indicating the day on which the game was played. It is an offset from the first day in the corresponding season. WTeamID is the team number for the winner team and WScore is the score for the winner team. For WLoc, it indicates the "position" of the victorious team. If the victorious team is the host team, the value will be "H". On the other hand, if the victorious team is the guest (or "away") team, the value will be "A". NumOT shows the quantity of additional periods played in the game, which is an integer equal to or greater than 0.

Fig. 2 illustrates the win ratio distribution for all the teams. The win ratio can be calculated as follows:

$$WinRatio = \frac{WinCount}{WinCoun + LossCount} \quad (1)$$

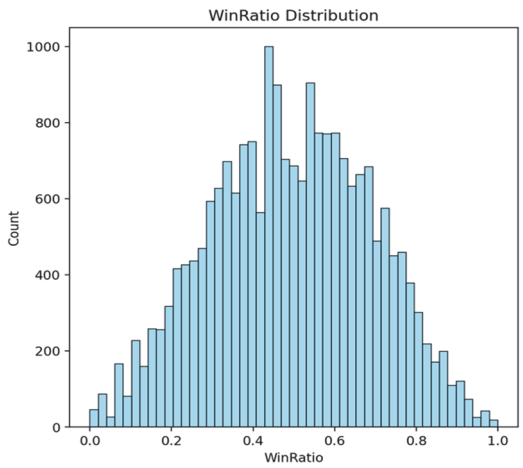

Figure 2 Win Ratio distribution. The distribution is similar to gaussian distribution and most of teams have a win ratio around 0.5

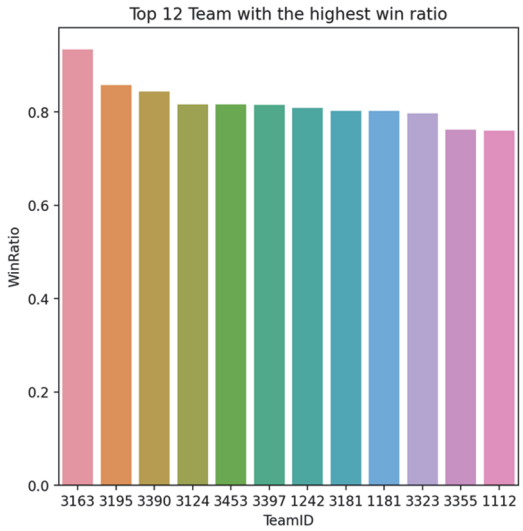

Figure 3 Top 12 team with the highest win ratio

From the historical season output, we summarize the win ratio for each team and give a team win ratio leader board as shown in Fig. 3. The top 12 teams with the highest win ratio are 3163, 3195, 3390, 3124, 3453, 3397, 1242, 3181, 1181, 3323, 3355, 1112. The team 3163 has a highest win ratio of 0.934 and a win number of 773 over 882 NCCA games.

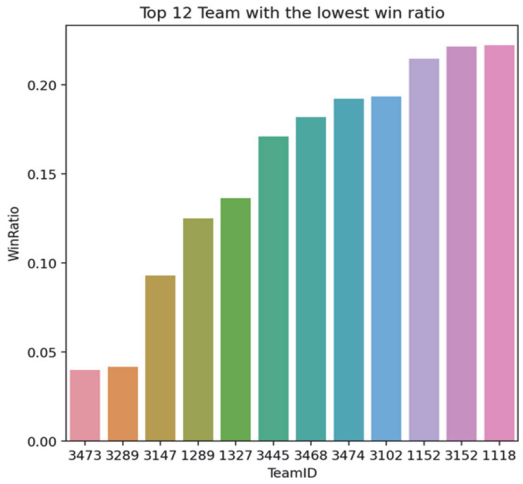

Figure 4 Top 12 team with the lowest win ratio

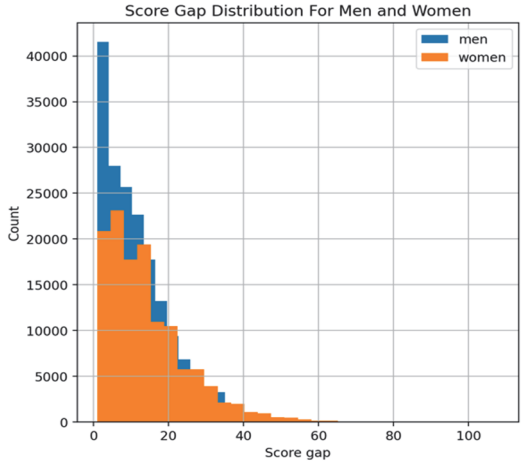

Figure 5 Score gap distribution for men and women

The team ID of the lowest score is 3473 which has the lowest win ratio of 0.04 as shown in Fig. 4. The win ratio is a derived feature and it is an important feature for the model training.

Since the competitors include women and men, it is essential to check the data distribution like score gap between men and women. From this distribution, it can be observed that it is not a uniform distribution. In most cases, the smaller the score gap, the more frequent the corresponding match occurrences. Additionally, the score gap for the male team is smaller than the score gap for the female team, and the distribution plot for females exhibits a longer tail.

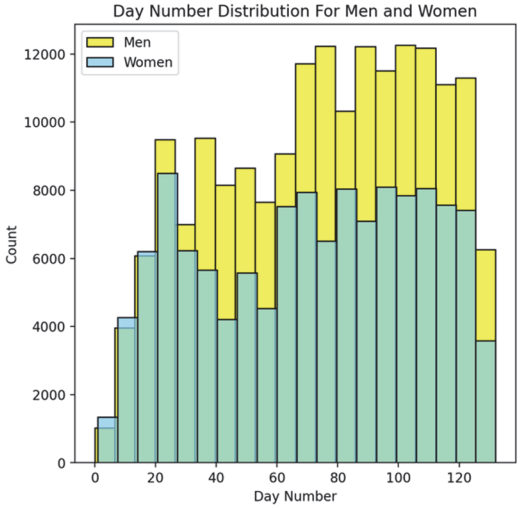

Figure 6 Day number distribution

Fig. 6 shows the day number distribution for both men and women. From the graph, it is obvious that there are more matches for males compared to females. Additionally, the NCCA matches for males are more concentrated in the latter part, especially in the interval of Day Numbers from 70 to 110.

3 DATA PREPARATION

In this section, the data preparation and feature engineering are introduced. The data preparation includes data collection and data cleaning part. We did not choose all data for training; instead we filter the data start from 2016 because of the FiveThirtyEight ratings emerging. Another reason is that the prediction is only for the 2023 NCAA games and some historical game data lack reference value.

Feature engineering is the process of transforming and selecting relevant data features to improve model performance and accuracy in machine learning. It involves creating new features, handling missing data, scaling, and selecting the most informative variables for better predictions. As shown in Tab. 1, the features like winRatio and GapAvg are aggregate features based on historical data from NCAA basketball games. In most cases, a higher winRatio and a higher GapAvg means a team has a better performance. The difference features like SeedDiff, GapAvgDiff and winRatioDiff are calculated by the difference between the same features of the two teams. A higher difference score means that team A has a greater advantage over team B in basketball skills.

Table 1 Features and Meanings for NCAA basketball competitions

Features	Meanings
SeedA	Seed Number for team A
SeedB	Seed Number for team B
Sex	Indicates whether the game is for women and men
WinRatioA	Win ratio for team A
WinRatioB	Win ratio for team B
GapAvgA	Gap between A's average score and all teams' average score
GapAvgB	Gap between B's average score and all teams' average score
SeedDiff	Seed Number difference between team A and team B
GapAvgDiff	GapAvg score difference between team A and team B
WinRatioDiff	Win Ratio difference between team A and team B

4 ALGORITHMS AND PRINCIPLE

4.1 Support Vector Machine

Support Vector Machine (SVM) [32] is a powerful supervised learning algorithm used for classification and regression tasks. It is widely used in the field of machine learning and data mining due to its ability to handle both linear and non-linear data separation. The primary goal of SVM is to find the optimal hyperplane that best separates different classes in the feature space. The hyperplane is chosen in such a way that it maximizes the margin between the closest data points from each class, known as support vectors. This property allows SVM to be robust and effective in dealing with noisy and overlapping data. SVM can handle linearly separable data efficiently, but it can also handle non-linear data through the use of kernel functions. These kernel functions map the original data into a higher-dimensional feature space, where the data becomes linearly separable, enabling SVM to solve complex classification problems.

In this work, the soft-margin SVM classifier is used for the hybrid model. The objective of this classifier is to minimize the following loss:

$$\left[\frac{1}{n}\sum_{1=1}^{n}\max\left(0,1-y_i\left(W^{Tn}x_i-b\right)\right)\right]+\lambda\|W\|^2 \tag{2}$$

4.2 Gradient Boosting Decision Tree

Gradient Boosting Decision Tree (GBDT) is a popular machine learning algorithm known for its strong predictive capabilities. It belongs to the ensemble learning family and is widely used for both classification and regression tasks. GBDT works by combining multiple weak learners, typically decision trees, into a strong predictive model. The algorithm builds these trees sequentially, with each new tree focusing on correcting the errors made by its predecessors. This process is achieved by placing higher weights on the misclassified data points during training. During the training process, GBDT employs gradient descent optimization to minimize the loss function, which measures the difference between the predicted and actual values. The gradients guide the algorithm to find the optimal split points in each decision tree, leading to improved model performance with each iteration.

At the m-th step of generic gradient boosting, a decision tree, denoted as $h_m(x)$, is fitted to pseudo residuals. The tree has a total of J_m leaves, which partition

the input space into J_m disjoint regions, denoted as R_{1m}, …, R_{Jm}. In each region, the tree predicts a constant value. Using indicator notation, the output of $h_m(x)$, for input x can be expressed as the sum:

$$h_m(x) = \sum_{j=1}^{J_m} b_{jm} I_{R_{jm}}(x) \tag{3}$$

where b_{jm} is the value predicted in the region R_{jm}. Next, the coefficients b_{jm} are multiplied by a factor γ_m, which is determined through a line search process aimed at minimizing the loss function. After this step, the model is updated as follows:

$$\begin{aligned} F_m(x) &= F_{m-1}(x) + \gamma_m h_m(x), \\ \gamma_m &= \arg\min_{\gamma} \sum_{i=1}^{n} L\left(y_i, F_{m-1}(x_i)\right) + \gamma h_m(x_i) \end{aligned} \tag{4}$$

Friedman suggests a modification to the algorithm, where it selects a distinct optimal value γ_m for each region of the tree, instead of using a single γ_m for the entire tree. He names this adapted algorithm "TreeBoost." In this case, the coefficients b_{jm} obtained from the tree-fitting procedure can be discarded, and the model update rule becomes:

$$F_m(x) = F_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I_{R_{jm}}(x) \tag{5}$$

$$\gamma_{jm} = \arg\min_{\gamma} \sum_{x \in R_{jm}} L\left(y_i, F_{m-1}(x_i) + \gamma\right) \tag{6}$$

4.3 TabNet

TabNet is a cutting-edge deep learning model that has gained significant attention for its effectiveness in tabular data processing and predictive tasks. Introduced in the research paper "TabNet: Attentive Interpretable Tabular Learning," TabNet combines ideas from both attention mechanisms and decision trees to achieve superior performance on tabular data, such as structured datasets commonly found in data science and machine learning projects.

The key innovation of TabNet lies in its use of a sequential attention mechanism, which learns to focus on relevant features during each decision step. Unlike traditional decision tree-based models, TabNet uses the Gated Linear Units (GLU) as a gating mechanism. This enables the model to make selective decisions based on the input features' importance.

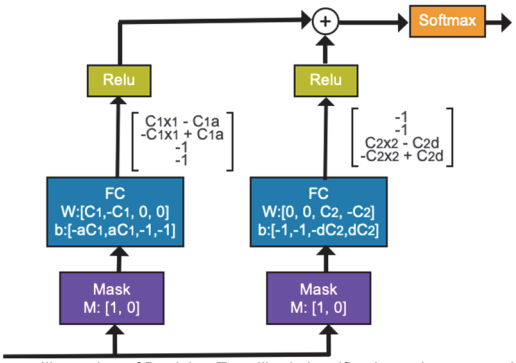

Figure 7 Illustration of Decision Tree liked classification using conventional Deep Neural Network (DNN) blocks

Decision trees liked algorithms like Xgboost and Lightgbm have made great success for tabular datasets. Fig. 7 shows a design how to make conventional DNN achieve the decision boundaries in hyperplane form. TabNet is built upon these principles and surpasses traditional decision trees (DTs) while retaining their advantages through meticulous design that includes: (i) leveraging sparse instance-wise feature selection learned from the data; (ii) creating a sequential multi-step architecture, where each step contributes to a fraction of the decision-making process based on the selected features; (iii) enhancing learning capacity through non-linear processing of the chosen features; and (iv) emulating ensemble learning [33] through higher dimensions and multiple steps.

For the feature transformation, "Feature transformer" is the key component which is designed as two sequential parts, as shown in the network structure diagram (Fig. 8) below:

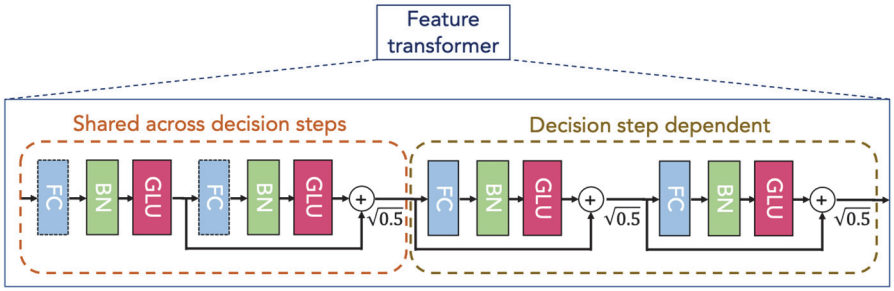

Figure 8 A feature transformer block example

Shared across decision steps: As the name suggests, these layers are shared across each decision step, meaning they are common across all steps.

Decision step dependent: These layers are unique to each decision step. They are separate layers that require individual training for each step.

As shown in Fig. 9, the Attentive Transformer (AT) is responsible for feature selection at each step. Feature selection is achieved by applying the sparse max activation (instead of GLU) while simultaneously considering prior proportions. The prior proportions allow us to control the frequency of the model selecting a feature and are controlled by the frequency it has been used in preceding steps.

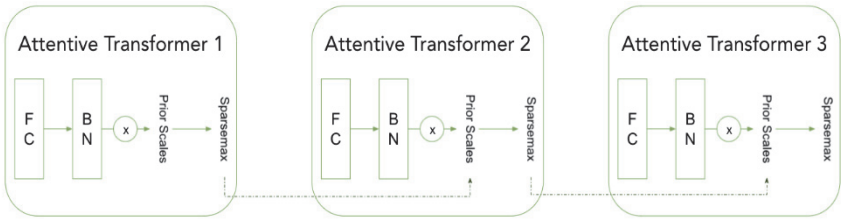

Figure 9 Attentive transformer block

To get the output of the TabNet, non-linear transformation followed by fully connected layer mapping. The final output of TabNet involves applying ReLU transformation to the outputs of Feature Transformers in each step. The outputs from all steps are then summed, passed through a fully connected layer, resulting in the ultimate output.

4.4 Proposed Model

The proposed model is a hybrid machine learning model, which is made up of support vector machine, gradient boosting decision tree and Tabnet. Fig. 10 shows the flow chart of the proposed model. After feature processing, the features are as input into each model for inference respectively.

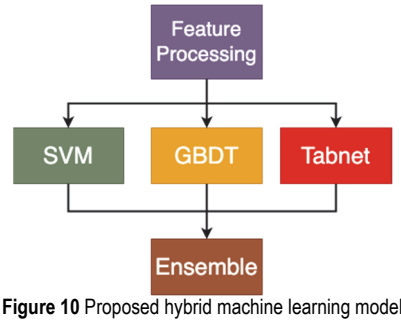

Figure 10 Proposed hybrid machine learning model

The corresponding output for each model is the winner probability of the team A versus team B. There are many common machine learning model ensemble methods [34]: Voting Ensembles, Bagging, Stacking and Weighted Average. In this paper, we choose the weighted average as the ensemble algorithm. It can combine predictions by assigning different weights to models based on their performance. Normally, the high performance is desired to have a higher weight compared with other low performance models.

5 EXPERIMENTS AND EVALUATION

5.1 Evaluation Metric

The accuracy score and Brier score are used for model evaluation. The Brier Score is a statistical metric used to evaluate the accuracy of probabilistic predictions. It measures the mean squared difference between predicted probabilities and actual outcomes, offering insight into the calibration and reliability of prediction models. Commonly applied to binary or categorical events, the Brier Score quantifies the precision of a predictive model by assessing how well it aligns with real-world results.

In practice, lower Brier Scores indicate better-calibrated predictions, where values closer to zero signify more accurate prediction. The Brier Score is particularly valuable for assessing the performance of probabilistic prediction, making it a staple in fields like meteorology, finance, epidemiology, and sports analytics. It provides a robust assessment of prediction quality, highlighting areas where predictions might be overly confident or excessively conservative.

The Brier score is commonly expressed as:

$$BS = \frac{1}{N}\sum_{t=1}^{N}\left(f_t - o_t\right)^2 \tag{7}$$

where f_t represents the prediction probability, o_t stands for the actual event outcome at time instance t (0 if it does not occur, and 1 if it happens), and N represents the total number of prediction instances.

5.2 Experiments

Similar to other neural network models, the TabNet algorithm exhibits high sensitivity to its hyperparameters. Therefore, the careful adjustment and selection of appropriate hyperparameters hold significant importance. In general, crucial parameters influencing the TabNet's performance encompass N_steps, feature_dim, gamma, lambda_sparsity, and others. The meaning of the parameters is shown in the following context:

N_steps: Number of decision steps, which controls the quantity of generated attention masks. Larger values offer more choices but can also increase computational cost.

Feature_dim: Determines the output dimension of the self-attention layers. Larger values might increase model representation power but also the risk of overfitting.

gamma: Controls the strength of the sparsity regularization term, influencing the degree of feature selection in the model.

Lambda_sparse: Weight of the sparsity regularization term, used to balance feature selection and predictive performance.

output_dim: Output dimension of the model, typically corresponding to the number of classes in the task.

Table 2 Specific setting of parameters

Settings	Values
N_steps	3
Feature_dim	12
Output_dim	2
Gamma	1
Lambda-sparsity	1e-3
Batch_ size	128
Epoch	140
Learning rate	0.02
Optimizer	Adam
Mask_type	entmax

Achieving a balance between performance and complexity is best attained through fine-tuning

feature_dim and output_dim values.

This paper optimizes and selects model parameters, as displayed in Tab. 2, through parameters search algorithm like grid search or bayes search to find the best parameters.

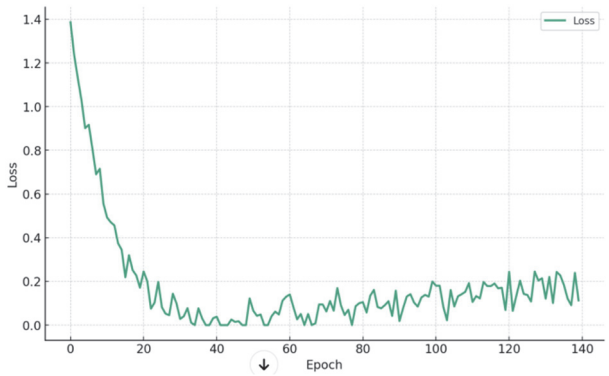

Figure 11 Training loss cure for TabNet

Fig. 11 shows the training progress for Tabnet model, the model is suitable for this task and begins to converge around 40 epoch.

Tab. 3 shows the comparison of Brier score versus different models which is introduced in the previous sections.

Table 3 Brier score of models

Model	Brier score
Support vector machine	0.181
Gradient Boost Decision Tree	0.185
Tabnet	0.184
Proposed Model	0.176

Table 4 Accuracy score of models

Model	Accuracy score
Support vector machine	0.647
Gradient Boost Decision Tree	0.632
Tabnet	0.643
Proposed Model	0.659

Tab. 3 gives the Brier score of different models and shows the superiority of the proposed model. Compared to the other singular model, the proposed model has a minimum brier score of 0.174 among all the models. The lower brier score means a better prediction accuracy in terms of the NCAA basket ball outcome. The proposed model is an ensemble model with a weighted average ensemble method.

The accuracy experiment shown in Tab. 4 also displays the power of the proposed method. Among all the methods, proposed hybrid model achieve a best score for prediction.

For the weighted averaging algorithm, a better model should have a higher weight, and the sum of weights for all models needs to be equal to 1. The weights are 0.5, 0.25, 0.25 respectively for SVM, GBDT and tabnet. The SVM contributes to a higher weight due to a higher performance than other two models.

The feature importance is a significant factor to decide which feature is better and which feature should remain. Hence, it is often used for feature selection. The attention mechanism in TabNet allows the model to automatically focus on the crucial parts of the input features. From the attention weights learned by the model in various tasks, it is easy to infer which features play a more significant role in the model's predictions. TabNet's feature importance is a byproduct of its attention-based decision-making process. It reflects how much each feature contributes to the model's decisions across the entire decision-making sequence, providing insights into which features are driving the model's predictions. The following picture shows the feature importance of Tabnet in a descending order. The SeedB is the most important feature while the GapAvgDiff is the least important feature with an importance of below 0.06.

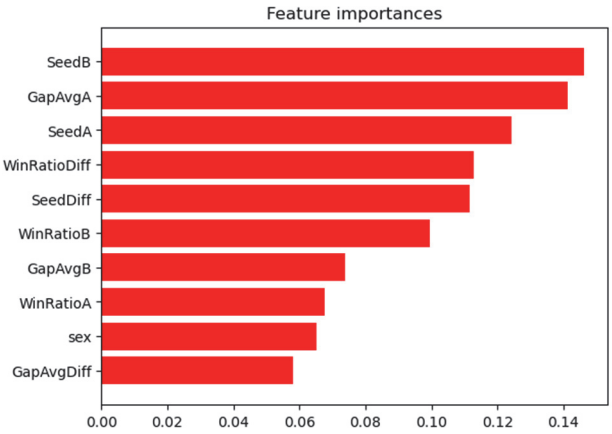

Figure 12 Feature importance of Tabnet

6 LIMITATION AND FUTURE WORK

Although our method has achieved good results in NCAA basketball predictions, it may not necessarily produce similar outcomes in other tasks, which is largely related to feature engineering. Moreover, the lack of additional modal information inputs, such as images, voice, text, or video, can to some extent affect the predictive performance of game outcomes. In the future, it is important to explore more kinds of sport games and investigate multi-modal models to improve the prediction result.

7 CONCLUSIONS

This paper presents a hybrid model combining Support Vector Machine, TabNet, and Gradient Boost Decision Tree for predicting NCAA game outcomes using tabular data. Through data exploration, we reveal insightful visualizations, such as win ratio distributions and score comparisons across genders, aiding in understanding and feature engineering based on attribute significance. The engineering process derives key features for model training. Experimentally, we detail TabNet's parameters (N_steps, gamma, learning rate) and demonstrate the hybrid model's superior Brier score performance compared to individual models like GBDT, SVM, and TabNet. The synergy of these models in the hybrid framework enhances prediction accuracy by merging their strengths and mitigating their limitations. Lastly, we analyze feature importance to guide further feature engineering efforts.

Acknowledgements

Thanks to the NCAA and Kaggle platform for providing datasets and data explanation. In addition, Cui

Kaiwen and Li Xuanyi have done the data preparation and feature processing for NCAA dataset. Cui Kaiwen also did the implementation for modelling of SVM, GBDT, Tabnet and Proposed model. Finally, we thank to the Yang Shuo for parameters searching for different model and data visualization in the feature engineering part.

8 REFERENCES

[1] Wikipedia contributors. (2024). *National Collegiate Athletic Association*. In Wikipedia, The Free Encyclopedia.

[2] Sonas, J. M. & Cukierski, W. (2023). *March Machine Learning Mania 2023*. Kaggle.

[3] Liu, X., Zhang, Z., Hao, Y., Zhao, H., & Yang, Y. (2024). Optimized OTSU Segmentation Algorithm-Based Temperature Feature Extraction Method for Infrared Images of Electrical Equipment. *Sensors, 24*(4), 1126. https://doi.org/10.3390/s24041126

[4] Zhao, H., Zhang, Z., Yang, Y., Gan, P., & Liu, X. (2023). Real-time reconstruction of temperature field for cable joints based on inverse analysis. *International Journal of Electrical Power & Energy Systems, 144*, 108573. https://doi.org/10.1016/j.ijepes.2022.108573

[5] Zhao, H., Zhang, Z., Yang, Y., Xiao, J., & Chen, J. (2023). A Dynamic Monitoring Method of Temperature Distribution for Cable Joints Based on Thermal Knowledge and Conditional Generative Adversarial Network. *IEEE Transactions on Instrumentation and Measurement, 72*, 1-14, 4507014. https://doi.org/10.1109/tim.2023.3317485

[6] Dan, Y., Zhang, Z., Gan, P., Ye, H., Li, Q., & Deng, J. (2020). Performance Analysis of Corroded Grounding Devices with an Accurate Corrosion Model. *CSEE Journal of Power and Energy Systems, 9*(3), 1235-1247. https://doi.org/10.17775/CSEEJPES.2020.03280

[7] Buyrukoğlu, S. & Savaş, S. (2023). Stacked-based ensemble machine learning model for positioning footballer. *Arabian Journal for Science and Engineering, 48*(2), 1371-1383.https://doi.org/10.1007/s13369-022-06857-8

[8] Huang, C. & Jiang, L. (2021). Data monitoring and sports injury prediction model based on embedded system and machine learning algorithm. *Microprocessors and Microsystems, 81,* 103654. https://doi.org/10.1016/j.micpro.2020.103654

[9] Naglah, A., Khalifa, F., Mahmoud, A., Ghazal, M., Jones, P., Murray, T., & El-Baz, A. (2018, December). Athlete-customized injury prediction using training load statistical records and machine learning. *2018 IEEE international symposium on signal processing and information technology (ISSPIT)*, 459-464. https://doi.org/10.1109/ISSPIT.2018.8642739

[10] Min, B., Kim, J., Choe, C., Eom, H., & McKay, R. B. (2008). A compound framework for sports results prediction: A football case study. *Knowledge-Based Systems, 21*(7), 551-562. https://doi.org/10.1016/j.knosys.2008.03.016

[11] Thabtah, F., Zhang, L., & Abdelhamid, N. (2019). NBA game result prediction using feature analysis and machine learning. *Annals of Data Science, 6*(1), 103-116. https://doi.org/10.1007/s40745-018-00189-x

[12] Sharma, M., Monika, K. N., & Pardeep, K. (2021). Naive bayes-correlation based feature weighting technique for sports match result prediction. *Evolutionary Intelligence*, 1-16. https://doi.org/10.1007/s12065-021-00629-3

[13] Pisner, D. A. & Schnyer, D. M. (2020). Support vector machine. *Machine learning*, 101-121. https://doi.org/10.1016/B978-0-12-815739-8.00006-7

[14] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189-1232. https://doi.org/10.1214/aos/1013203451

[15] Turner, C. R., Fuggetta, A., Lavazza, L., & Wolf, A. L. (1999). A conceptual basis for feature engineering. *Journal of Systems and Software, 49*(1), 3-15. https://doi.org/10.1016/S0164-1212(99)00062-X

[16] Le Cun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436-444. https://doi.org/10.1038/nature14539

[17] Montavon, G., Samek, W., & Müller, K. R. (2018). Methods for interpreting and understanding deep neural networks. *Digital signal processing, 73*, 1-15. https://doi.org/10.1016/j.dsp.2017.10.011

[18] Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). *Deep learning for computer vision: A brief review. Computational intelligence and neuroscience*. https://doi.org/10.1155/2018/7068349

[19] Lauriola, I., Lavelli, A., & Aiolli, F. (2022). An introduction to deep learning in natural language processing: Models, techniques, and tools. *Neurocomputing, 470*, 443-456. https://doi.org/10.1016/j.neucom.2021.05.103

[20] Tan, M. & Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, 6105-6114.

[21] Zou, X., Yang, J., Zhang, H., Li, F., Li, L., Wang, J., & Lee, Y. J. (2024). Segment everything everywhere all at once. *Advances in Neural Information Processing Systems, 36*. https://doi.org/10.1038/s41467-024-44824-z

[22] Raschka, S., Liu, Y. H., Mirjalili, V., & Dzhulgakov, D. (2022). *Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python.* Packt Publishing Ltd.

[23] Pang, B., Nijkamp, E., & Wu, Y. N. (2020). Deep learning. with tensorflow: A review. *Journal of Educational and Behavioral Statistics, 45*(2), 227-248. https://doi.org/10.3102/1076998619872761

[24] Devlin, J., Chang, M. W., Lee, K., et al. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805, 2018.*

[25] Radford, A., Wu, J., Child, R., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog, 1*(8), 9.

[26] Fraiwan, M. & Khasawneh, N. (2023). A Review of ChatGPT. Applications in Education, Marketing, Software Engineering, and Healthcare: Benefits, Drawbacks, and Research Directions. *arXiv preprint arXiv:2305.00237, 2023*.

[27] Arik, S. Ö. & Pfister, T. (2021). Tabnet: Attentive. interpretable tabular learning. *Proceedings of the AAAI conference on artificial intelligence, 35*(8), 6679-6687. https://doi.org/10.1609/aaai.v35i8.16826

[28] Zhuohan, W. (2022, December). Tabnet With Data Augmentation Apporach in Stock Return Prediction Task. *2022 19th International Computer Conference on Wavelet Active Media Technology and Information Processing* (ICCWAMTIP), 1-5. https://doi.org/10.1109/ICCWAMTIP56608.2022.10016580

[29] Cai, Q. & He, J. (2022). Credit Payment Fraud detection model based on TabNet and Xgboot. *2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, 823-826. https://doi.org/10.1109/ICCECE54139.2022.9712842

[30] Nguyen, H. V. & Byeon, H. (2023). Predicting Depression during the COVID-19 Pandemic Using Interpretable TabNet: A Case Study in South Korea. *Mathematics, 11*(14), 3145. https://doi.org/10.3390/math11143145

[31] Yan, J., Xu, T., Yu, Y., & Xu, H. (2021). Rainfall forecast model based on the tabnet model. *Water, 13*(9), 1272. https://doi.org/10.3390/w13091272

[32] Steinwart, I. & Christmann, A. (2008). Support vector machines. *Springer Science & Business Media*. https://doi.org/10.1007/978-0-387-77242-4

[33] Sagi, O. & Rokach, L. (2018). Ensemble learning: A survey. *Wiley interdisciplinary reviews: data mining and knowledge discovery, 8*(4), e1249.https://doi.org/10.1002/widm.1249
[34] Dietterich, T. G. (2000). Ensemble methods in machine learning. *International workshop on multiple classifier systems*, 1-15. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-45014-9_1

Contact information:

Kaiwen CUI
(Corresponding author)
The Second High School Attached to Beijing Normal University,
International Division,
Beijing, China
E-mail: cuikaiwen_ai@163.com

Xuanyi LI
The Affiliated High School of Peking University's Dalton Academy,
Beijing, China
E-mail: lixuanyi2025@i.pkuschool.edu.can

Shuo YANG
The Second High School Attached to Beijing Normal University,
International Division,
Beijing, China
E-mail: yangshuo250@outlook.com