IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 October 2023, accepted 16 November 2023, date of publication 20 November 2023,
date of current version 27 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3334911

APPLIED RESEARCH

Learning Performance Efficiency of College Basketball Players Using TVAE

NASIM YAHYASOLTANI[1], (Member, IEEE), **PRIYANKA ANNAPUREDDY**[1], AND **MANZUR FARAZI**[2]
[1]Department of Computer Science, Marquette University, Milwaukee, WI 53233, USA
[2]Department of Mathematical and Statistical Sciences, Marquette University, Milwaukee, WI 53233, USA

Corresponding author: Nasim Yahyasoltani (nasim.yahyasoltani@marquette.edu)

This work was supported by the Northwestern Mutual Data Science Institute (NMDSI) and the Athletic and Human Performance Research Center (AHPRC) Grants.

ABSTRACT Monitoring workload is essential to evaluate possibility of fatigue or injuries and overall performance of the players. Large player's workload in any game including basketball can contribute to the stress and fatigue and overloaded players may get exhausted and exhibit burnout symptoms, which eventually result in their lower efficiency in the game. This paper aims at predicting the efficiency of the basketball players in all positions (guard, forward and center) based on their workload information. Machine learning (ML) methods for regression and classification are applied to the dataset for predictive modelling of the variables. The analysis includes: (i) one-model for all player positions and (ii) position-based models for respective player position. Leveraging tabular variational autoencoders (TVAE), synthetic data is generated to improve the accuracy. The evaluated models from both regression and classification verify that better accuracies can be obtained in the position-based models rather than the one-model for all positions approach. The performance analysis of the algorithms indicates that the player's efficiency can be estimated from the workload information which can provide valuable insight and individualized recommendation for optimal performance during the competition.

INDEX TERMS Basketball, classification, player efficiency, PCA, regression, tabular variational autoencoders, workload.

I. INTRODUCTION

Workloads are defined as cumulative stresses on athlete in training sessions [1]. Wearable sensors using global positioning system (GPS) facilitate accurate measurements of some of the workload variables. Studying college men's basketball players workload and game performance gives coaches and researchers an idea of how players respond over time with training, how the workloads influence their performances during competitive games and how workloads can be increased optimally to get best performances. Additionally, the study of workloads variables with the connection of players' game performance is extremely important to set individualized training schedules for the players. It gives an overall idea in a feature selection framework to select the most important workload variables.

II. LITERATURE REVIEW

There are studies to understand the during-competition stress and demands and the comparison of the workloads during training with workloads during actual com-petition [2]. Numerous studies have aimed to investigate the significance of workloads during competition and how these workloads vary across different situations. The objective of these studies is to understand how workloads fluctuate under various circumstances. Average PlayerLoad, PlayerLoad per minute (PL/min), high inertial movement analysis (high-IMA), and jumps were compared by (a) season, (b) player position, and (c) game outcome (wins vs. losses) using linear mixed model [3]. They concluded that when jump was increased over the

The associate editor coordinating the review of this manuscript and approving it for publication was Sangsoon Lim.

time (season), PL/min was found higher in guard position, and high-IMA was lower in the win games. In [4] external training load was compared with the internal training load, expressed by the session rating of perceived exertion (sRPE) and significant correlation was observed between sRPE and the external load variables. The authors also observed strong correlation between Player Load (PL) and the total number of Acceleration (ACC), Deceleration (DEC), Changes of Direction (CoD), and jumps done within highband. In a similar study [5], training load (TL) profile, i.e., heart rate (HR), session rating of perceived exertion (sRPE) were examined during the crucial parts of the competitive season (pre–play-off finals) and significant correlation was observed between sRPE and all HR based. However, a recent study showed that higher training workloads do not correspond to the best performances of elite basketball [6]. In [7], the authors studied the relationship between internal and external load of women basketball players and found that load experienced during competition was significantly higher compared to training. Additionally, there were notable differences in load based on playing positions particularly between the backcourt and frontcourt players. Session distance was shown as the most important predictor of the rating of perceived exertion (RPE) using artificial neural network (ANN) and generalized estimating equation (GEE) models by comparing session distance, high-speed running (HSR) in professional Australian football (AF) players in [8]. It was also shown in [9] that internal and external TL variables were correlated with performance during international women's basketball games.

With recent advances in machine learning (ML) algorithms and the increase in computational power, there is more interest in the topic of basketball analytics where data-driven approaches are used to improve player and team performance. The influence of players' statistics in the outcome of the game were studied in [10], [11], and [12]. Visualization of basketball shot, and statistics was addressed in [13]. The movement data of the basketball player to identify his postures was incorporated in [14] to improve the coach's guidance. A system to determine the performance of the player in defense or offence was proposed in [15]. In [16] the authors do basketball shot analysis using ML algorithms with accelerator data.

However, none of the existing literature models the association of players work-load and efficiency. The current study proposes to build ML models that are predictive of players efficiency using their workload information. The proposed model can guide the coach on how different players may react to different training schedules. This can then help designing better and more efficient training protocols. Usually, ML models require large amounts of data to exhibit better predictive performance. In most of the cases, the available data is either too small or costly to acquire. In the current work, variational autoencoder (VAE) a class of generative models is used to augment the existing real data and generate realistic samples from the original dataset distribution. Generating additional samples using VAE has been used widely in different application domain including images, videos, music, and text [17], [18], [19], [20], [21], [22]. More specifically, since the data is tabular, TVAE (Tabular Variational Autoencoder) has been incorporated [23].

This work makes a significant contribution to the existing literature by modeling the relationship between workload and efficiency and by building position-based models, which have shown better performance compared to a single model for all positions. Additionally, the study introduces the use of a time-varying autoencoder to enhance the predictive performance of the models. By incorporating TVAE, the research aims to improve the accuracy and reliability of predicting efficiency based on workload information. The contributions of this work can be summarized as follows: (1) the study initially identifies the most influential features in college basketball players' workload data. These features are crucial in understanding the relationship between workload and efficiency; (2) the inherent correlations and associations between the identified features and the performance efficiency of the players are modeled and learned using various machine learning (ML) algorithms. This step helps establish a quantitative understanding of how workload impacts efficiency; (3) this research develops algorithms tailored to each player's position in the game. This approach recognizes that different positions may have distinct workload-efficiency relationships, and by considering position-specific models, the accuracy of predictions can be improved; and (4) to further enhance the performance of the algorithms, the study proposes the use of TVAE to generate synthetic data. By incorporating TVAE, the models can learn additional patterns and variations in the data, leading to improved predictive performance.

The rest of the paper is organized as follows: The data used for the study is shown in figure 1 and definitions of the workload variables are described in section III; Preprocessing of the workload variables and player scores are detailed in section IV; Synthetic data generation and ML methods used are provided in section V; Results and discussion can be seen in section VI.

III. DATA DESCRIPTION

The study consists of a total of 34 college basketball players from Marquette University basketball (MUBB) team with an average weight of 204.3 lbs. (standard deviation of 24.3), height of 77.4 inches (standard deviation of 3.7), and a body mass index (BMI) of 23.9 kg/m^2 (standard deviation of 1.5). All athletes played at least one game for MUBB during the session of 2016-2020. The players participated in Division I National Collegiate Athletic Association (NCAA) college basketball. NCAA college basketball is exceedingly competitive tournament with a lengthy and challenging season. Each of the teams plays about 30–35 regular season games (1–3 games per week) over a 5-month period from November to March. The training session starts usually from

June for the following season. Workloads were collected during the training session as well as match time.

Wearable devices were used to collect workload data. The Catapult OptimEye S5 was placed in a supportive harness and positioned on the back to collect the data before every game and training session. Using the Catapult devices data were collected from the basketball players in each game and training session and exported to a csv file. About as many as 1500 workload variables such as Total Player Load (TPL), Player Load per Session (PLS), Jump Load (JL), Inertia Movement Analysis (IMA), and so on were collected from 34 players over the 5-year period. The target variable, player efficiency, is calculated from each game for each player separately based on different game scores performed by the player. The game score for each game as well as player information such as height or weight and game information such as win/loss, venue, opponent, pace, tempo are also stored.

Four different data sources were used for this study. Therefore, some preprocessing was needed to merge the data into a single file to be ready for the analysis. The preprocessing steps involved are presented as follows below.

IV. PREPROCESSING

A. PREPROCESSING OF CATAPULT WORKLOAD DATA

The catapult workload data was generated through catapult devices and there are many irrelevant variables. There are workloads for game time at different sessions: pre-game session, post-game session and non-game day training session. In the catapult data, there are 131412 observations on 1500 variables. We have selected 21 important workload variables and the workloads only during the game time were included. Workloads of multiple sessions of a game are aggregated.

B. PREPROCESSING OF BOX SCORE

The box score has the statistics related to game such as different points and im-portant elements of the game. These statistics were used to calculate the efficiency of the player.

Using the formula followed by MUBB, we calculated efficiency of a player as:

$$\begin{aligned} EFF = {} & Points - (2FGA - 2FGM) \\ & - (3FGA - 3FGM) - (FTA - FTM) \\ & + 2 * Off + Def - PF + 2 * Asst - 2 * TO \\ & + Block + Steal \end{aligned} \tag{1}$$

where Points is the total scores scored by the player, 2FGA is the 2-points field goal attempted, 2FGM is the 2-points field goal made, 3FGA is the 3-points field goal attempted, 3FGM is the 3-points field goal made, FTA is the free throw attempted, FTM is the free throw made, Off is offensive rebound, Def is the defensive rebound, PF is the power forward, Asst is the assistance given by a player that leads a point, TO is the turn over (a turnover occurs when a team loses possession of the ball to the opposing team before a player takes a shot at their team's basket), a Block occurs when a defensive player legally deflects a field goal attempt from an offensive play-er to prevent a score, and a Steal occurs when a defensive player legally causes a turnover by his positive, aggressive action(s).

The equation in (1) may not reflect clear picture for comparing the performance of two players. Because, if player X plays for 20 minutes and player Y plays for 2 minutes then comparing their performances using EFF does not seem justifiable. Hence, player efficiency rate per minute (PER) is used for each player to compare their performance as follows:

$$PER = EFF/MP \tag{2}$$

where MP represents the minutes played by a player in a game. We get an efficiency and PER value for each player for a game.

C. MERGING EFFICIENCY AND LOAD

To merge load and efficiency, we used date as a decision factor. At any given date, there are a set of loads of some players as well as a set of efficiencies of the same players. So, loads and efficiencies will be merged according to the date for each player. The final file has 1329 rows and 33 features which will be used for the analysis. The variables in the final data set are Player Name, Player Height, Player Weight, Player Position, Game Season, Game Date, Opponent Team, Result (Output of a Game), Game Venue, Tempo, Pace, Player Efficiency (EFF), Player Efficiency per Minute (PER), Duration, Distance, Total Player Load, Average Player Load, Active Player Load, Repeated High Intensity Effort (RHIE) Total Bouts, RHIE Efforts Per Bout Minimum, RHIE Efforts Per Bout Maximum, RHIE Efforts Per Bout Average, Jump Minute, Jump Load, Peak Player Load, Total IMA, High IMA, Total High IMA, PL/High IMA. The variables listed above are the most important factors according to the coaches to better understand the player's training effectiveness and game performances. The player load is considered as the driving force to measure the training efficiency which is the square root of the sum of the squared rates of change in acceleration between each moment of a training session along the movement axes (x, y, and z). Player load is a scientific way of athlete monitoring process that can provide a quick summary of an athlete's work. RHIE is the ability to identify periods where athletes have put in repeated 'high intensity' velocity efforts without adequate recovery that provides feedback on the relative fitness of an athlete. IMA is a set of metrics that measures athlete micro movements and direction regardless of unit orientation. Player load/high IMA is another important feature that characterizes the high intensity movements relative to the player load. Pace describes number of possessions a team uses per game, and Tempo refers how fast a team plays on the offensive end of the floor. Pace and Tempo describe the strategy and impact of the opponent team that also contributes to the player efficiency.

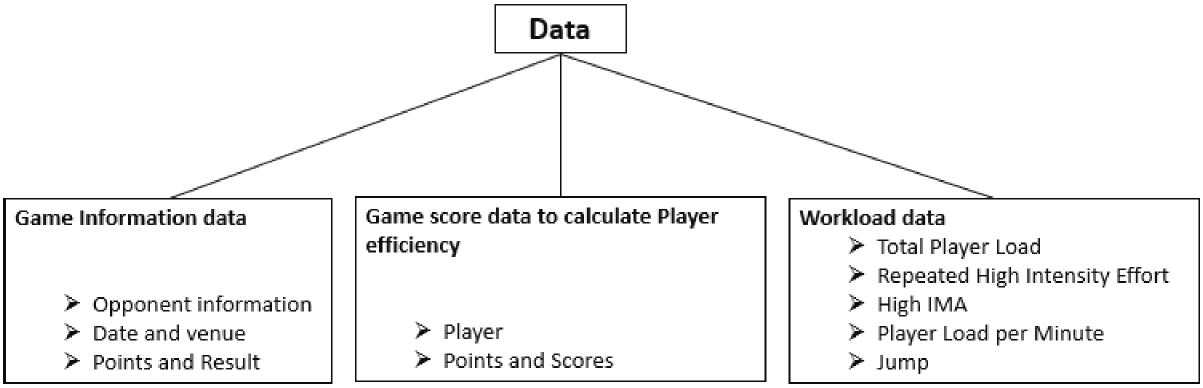


FIGURE 1. Data used in the study.

Among the selected variables, the numerical variables are normalized before analysis to minimize the effects due to unit differences.

D. DIVIDING DATA INTO SUBGROUPS

In basketball game, the role of different positioned players is not the same. Basically, there are three main positions in basketball as forward, center, and guard. Their duties and strategies are different. Hence, it is reasonable to analyze the player efficiency of these three groups separately. The range of PER for players in center position is quite different from that of players in guard and forward positions. To characterize the relationship between player PER and their workload information and how it changes with the player position, ML algorithms are employed separately for each player position. The models built on guard, forward and center positions are then compared with a one-model approach where a single model is built for all the positions.

E. SYNTHETIC DATA

Training ML models may be inefficient when the available data are either too small, or acquiring the labelled data is costly. Synthetic data generation is one of the effective solutions when there is not sufficient data. Generative adversarial network (GAN) [24] and VAE [23] are increasingly used to generate synthetics data in differ-ent applications where large amounts of data is required for effective modelling. The data distribution of the generated data can be controlled to have same statistical characteristics as that of the actual data. They are also effective in data imbalance applications where the availability of rare and infrequent patterns can be controlled. Generative models are extensively used to generate synthetic data from existing real data. As a class of generative models, VAE combines variational inference with deep learning (DL) and models the probabilistic distribution of the latent attributes in the encoder. The latent space is then sampled in the decoder to reconstruct the new (synthetic) data. The goal is the synthetic data is very similar to the real data. These generative models pose challenges with tabular data due to the discrete and continuous nature of the attributes; these are overcome in the VAE for tabular data, the so-called "TVAE" [23]. This work uses TVAE to generate synthetic data from the actual data of MUBB to increase the number of samples for training and improve the accuracy.

Chi–Squared (CS) and Kolmogorov-Smirnov (KS) tests can be used to compare the distribution of synthetic and original data for discrete and continuous columns. Detection metrics like logistic detection and svc detection can evaluate how hard it is to distinguish synthetic data from original data by a ML classifier [23], [24], [25], [26], [27], [28].

V. METHODS

This section details the methods used for predicting the PER of player using the workload variables. Initially, feature selection methods are applied to select the relevant workload variables and these workload variables are transformed into a new set of uncorrelated variables using principal component analysis (PCA) [29], [30], [31]. Supervised ML algorithms were used to build the predictive models using principal components as predictors. PER being a continuous variable initially regression methods are applied to predict PER. In order to improve the predictive performance, regression task has been converted into classification task. Therefore, column discretization has been performed on PER to enforce predicting PER into a classification problem. This is done by splitting the range of continuous attribute PER into nominal attribute PER group using the median value of PER. PER has been discretized into two groups: low performance and high-performance groups. Players with PER less than equal to the median value are labelled under low performance group otherwise high-performance group.

In the current work, both regression and classification methods are used to predict PER and PER group of the players respectively. PER group for classification models is based on the median of the PER of the players. Our study employed both one-model approach and position-based models to study the relationship between work-load variables and PER. In one-model approach, single model was used to predict player's PER irrespective of their position. Whereas in position-based models, separate models were built for each of the player positions guard, forward and center. Performance of the position-based models are then compared with that of one-model method. Opponent variables pace and tempo are also included to find their association with PER and if they could improve the predictive performance.

A. FEATURE SELECTION

The predictive performance of a model depends on the predictor attributes used and their association with the dependent variable. In this regard, to include the most relevant attributes in the model, feature selection using correlation analysis is exploited. Pearson's product moment correlation is applied on the workload and opponent variables to identify those that have strong impact in predicting the PER of a player. Strength of their association is represented by correlation coefficient (r). The correlation analysis is done separately for each player position to see how the association changes with the player position. A threshold of p-value=0.05 is considered for defining the statistical significance in this study. Higher correlation between workload variables and PER is observed in center position data with the variables JL, TPL having coefficients 0.46 and 0.44, respectively. Whereas in guard and forward position data, the coefficients ranged between 0.33-0.37 for these variables. The association of other variables RHIE, Dura, PLM, HIMA, APL, PPL, JM, THIMA, AcPL with PER also in-creased from guard to forward and then to center position which being the highest. HIMAPL of the players have higher impact on guard position player's PER compared to forward and center positions. The height of the players are seen to have no impact on player's PER in guard and forward positions but showed small factors of correlation with that of player's PER in center position.

B. SMALLER SET OF FEATURES WITH PCA

The features which have met the p-value criteria are included for the predictive analysis of PER. Correlation among the predictor variables is also evaluated to check for multicollinearity. It is observed that high relationships exist among predictor variables. To overcome the multicollinearity among the predictors, PCA was performed on these features blinded to PER. As a dimensionality reduction technique, PCA reduced the standardized correlated variables into new uncorrelated components retaining most of the variance in the first few components. 95% of the variability of the original variables is contained in the first 8 components. Therefore, the first eight principal components were selected as predictor variables for regression and classification tasks.

C. TABULAR VARIATIONAL AUTOENCODER

Let x be high-dimensional i.i.d samples drawn from the true data distribution p(x) over a random variable x. Generative modeling aims to learn from x to draw new samples such that they belong to the same distribution as p(x). Typically, VAEs encode the data samples x into a latent variable z via a probabilistic encoder, which is parameterized by a neural network. Then, a decoder is used to reconstruct the original input data based on the samples from z. The VAE maximizes the marginal likelihood of the reconstructed data. More specifically, $P(x|z)$ denotes a probabilistic decoder with a neural network to generate data x given the latent variable z, where $p(z)$ is a fixed prior distribution over latent space z. This posterior probability is often intractable and is usually approximated by a variational posterior where the parameters are learned through a neural network [32], [33], [34]. Since the real data is tabular, we need to incorporate VAE for tabular data, i.e., TVAE [23].

TVAE from sdv package [24] is used to generate synthetic data from the basketball data of the current study. Another 1000 and 10000 data points were generated separately using the TVAE model. As detailed in section III, the quality of the generated data is evaluated using statistical and detection metrics. It was observed from the evaluation metrics that generated synthetic data has a high similarity to the true data.

D. REGRESSION

Since PER is a continuous variable, it is natural to apply various regression models to describe how PER depends on the player workload information. Ordinary least squares (OLS), Ridge and Lasso regression techniques are applied to the data. OLS fits a linear model to the data and works towards minimizing the sum of squared errors [11], [12]. Ridge and Lasso are the regularization methods which add penalty to the coefficient estimates. The value of the regularization parameter λ is selected through cross-validation. Other regression techniques such as random forests regressor (RF), support vector regressor (SVR), K-Neighbors regressor (KNN), kernel Ridge and kernelized KNN [10] are applied to model the non-linear characteristics of the data. A single regression model for the full data; and 3 position-based regression models for guard, forward and center data are built and compared. These models are evaluated before and after adding the opponent variables. The models are evaluated using train-test split method. Algorithms were first trained on training dataset and the prediction models are generated. The predictive capabilities of the models are evaluated on unseen test dataset.

Root Means Squared Error (RMSE) and R-squared (R^2) are used as the evaluation metrics for regression models. RMSE indicates the absolute difference of how far the predicted values are from actual data points and, R^2 indicates the relative improvement of the model compared with mean model. Higher R^2 and lower RMSE provides better predictive performance [25].

E. CLASSIFICATION

In the classification task, predictive models are built for classifying players into low and high-performance groups. These groups (referred as PER group) are defined separately for each player position guard, forward and center; and for the full data based on the respective PER median values.

The median of PER for players in guard, forward and center positions is 0.39, 0.24 and 0.31 respectively; and 0.33 on the full data. If a player's PER is less than the respective median, the player is labelled as low performance otherwise high performance. Like regression, one-model approach and position-based models are built and evaluated with and without opponent variables to find the best performing model for predicting PER group using the trans-formed workload variables as predictors. Train-test split method is used for training and testing the algorithms.

For training the workload dataset, initially Logistic Regression (LR) has been ap-plied. LR is a probabilistic model which maximizes the likelihood to fit the model. Other classification algorithms like support vector classifier (SVC) and K-nearest neighbors (KNN) are applied to the data. Typically, SVC works well on the data where the classes are not linearly separable, by projecting the feature space into higher dimension. On the other hand, KNN is a technique which classifies the data based on the majority voting of the closest training data points. Bagging and boosting based algorithms like Random Forests (RF), AdaBoost and XGBoost are also applied on the data. These are the ensemble methods which improve the performance of weak learners through resampling techniques. Prediction Accuracy which is the percent-age of true predicted labels is used as the evaluation criteria for comparing the performance of these classification algorithms [30].

VI. RESULTS

A. REGRESSION AND CLASSIFICATION RESULTS ON ORIGINAL DATA

1) REGRESSION

Initially a single model was built for all the positions. Table 1 shows the predictive performance of the various regression algorithms employed in this approach before and after including the opponent variables. Given the non-linearity of the data, KNN and kernelized algorithms exhibited better performance. SVR and KNN have similar performances with comparatively higher R^2 of 0.17 and RMSE 0.33 before including opponent variables. When the opponent variables were added to these models, the performance improved with R^2 of 0.18 and lower RMSE 0.31 for the KNN model. Kernelized KNN also returned similar results after adding the opponent information. Both KNN and kernelized KNN were better performers with RMSE being low and R^2 high after adding the opponent information.

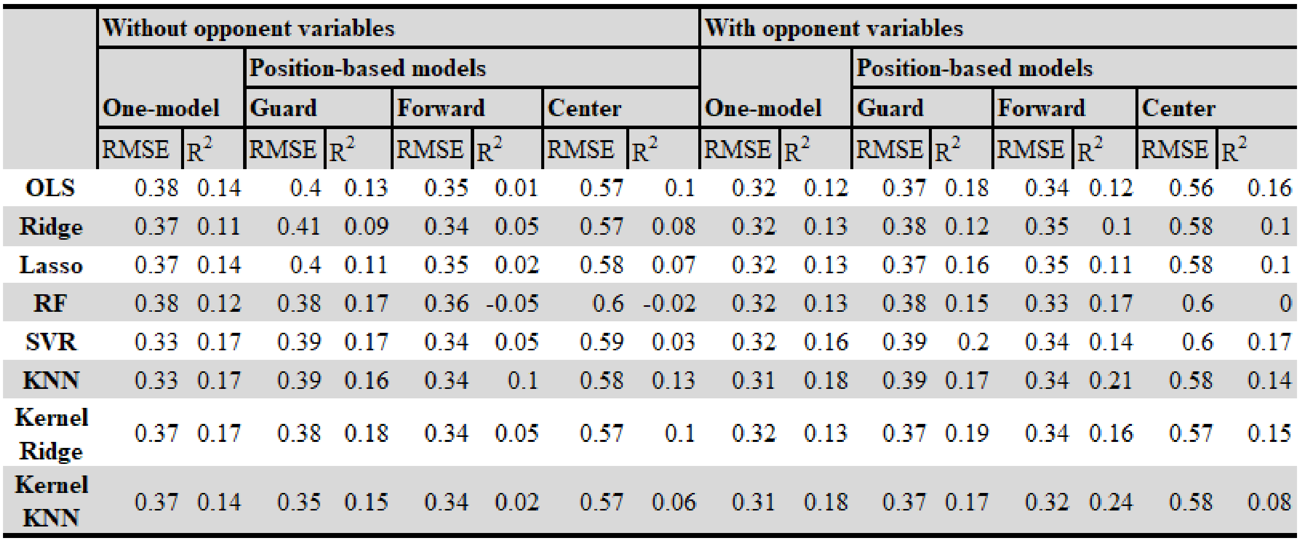
Table 1 also includes the comparison of evaluation results of position-based models for guard, forward and center positions before adding and after adding the opponent variables. Among the models without the opponent variables, R^2 is the highest as 0.18 with kernel ridge regression for guard position data. Relatively low performance was observed for forward and center positions with 0.10 and 0.13 R^2 respectively with KNN. After adding the opponent information to the models, R^2 of the KNN improved to 0.21 for forward position, and kernelized KNN has higher R^2 score of 0.24 for this data. The performance of guard and center position models also improved after adding the opponent information. SVR showed better performance with these positions with R^2 score of 0.20 and 0.17 for guard and center positions respectively. From the results, it can be concluded that KNN consistently performed well with forward position data, While SVR and kernel ridge performed with guard position.

Comparing the results of one-model and position-based regression approaches, it can be observed that position-based models have better performance in terms of R^2 than a one-model for full data approach. However, RMSE appears to be higher in these models. Opponent variables pace and tempo improved the performance of one-model and position-based models, their performance was lower without the opponent variables.

2) CLASSIFICATION

For predicting the PER group of players based on workload variables, different classification techniques have been applied on the whole dataset as well as the position-based cases. Table 2 shows the comparison of these two approaches without and with using the opponent information. Given the nonlinearity of the data, various algorithms like logistic regression (LR), support vector classifier (SVC) and K-nearest neighbors (KNN) are applied to the workload data. Other boosting and bagging based classifiers XGBoost (XGB), AdaBoost (ADA), and random forests (RF) are also learned on the training data. When these classifiers are evaluated on test set, LR, SVC, KNN, ADA and RF performed equally well (64.2%, 63.2%, 63.8%, 63.3%, and 63.6%) in the one-model approach without the opponent variable. The predictive accuracy of LR 64.2% is the highest among them. When the opponent information is added, the performance of the models slightly dropped with 63.6% being the highest with KNN. Table 2 also includes the performance of position-based classification models for each of the player positions guard, forward and center. Among these models, the predictive accuracy is high for the center position classifier compared to the other classifiers (guard and forward) with and without opponent information. This is as expected from the correlation observed between workload variables and PER for center position. Next to the center position, better performance is seen with forward position and then relatively low with guard position models. Similar to the one-model approach, the position-based classifiers accuracy is comparatively high without the opponent information added. KNN is seen to be the best performer of center and forward positions with 71.2% and 64% respectively without using opponent information. After adding the opponent variables, the KNN model returned nearly 69% and 60% for these positions respectively. The other algorithms SVC, KNN, ADA and RF have relatively low performance with opponent-based models. Therefore, unlike the regression models, performance of classification models is observed to be better

TABLE 1. Regression models performance.



	Without opponent variables								With opponent variables							
			Position-based models								Position-based models					
	One-model		Guard		Forward		Center		One-model		Guard		Forward		Center	
	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2
OLS	0.38	0.14	0.4	0.13	0.35	0.01	0.57	0.1	0.32	0.12	0.37	0.18	0.34	0.12	0.56	0.16
Ridge	0.37	0.11	0.41	0.09	0.34	0.05	0.57	0.08	0.32	0.13	0.38	0.12	0.35	0.1	0.58	0.1
Lasso	0.37	0.14	0.4	0.11	0.35	0.02	0.58	0.07	0.32	0.13	0.37	0.16	0.35	0.11	0.58	0.1
RF	0.38	0.12	0.38	0.17	0.36	-0.05	0.6	-0.02	0.32	0.13	0.38	0.15	0.33	0.17	0.6	0
SVR	0.33	0.17	0.39	0.17	0.34	0.05	0.59	0.03	0.32	0.16	0.39	0.2	0.34	0.14	0.6	0.17
KNN	0.33	0.17	0.39	0.16	0.34	0.1	0.58	0.13	0.31	0.18	0.39	0.17	0.34	0.21	0.58	0.14
Kernel Ridge	0.37	0.17	0.38	0.18	0.34	0.05	0.57	0.1	0.32	0.13	0.37	0.19	0.34	0.16	0.57	0.15
Kernel KNN	0.37	0.14	0.35	0.15	0.34	0.02	0.57	0.06	0.31	0.18	0.37	0.17	0.32	0.24	0.58	0.08

without the opponent variable added. Similar trend is seen with guard position models. LR returned better performance compared to other algorithms for guard position without and with opponent information 59% and 58% respectively.

From the analysis of the regression and classification results, it is observed that, the use of position-based models showed improved performance compared to a single model applied to all players. This suggests that considering specific characteristics and workload patterns of each player's position can enhance prediction accuracy. The inclusion of opponent information in the regression models led to improved performance. This indicates that incorporating the opposing team's characteristics or playing style can provide valuable insights into a player's efficiency and performance. Surprisingly, the addition of opponent information did not contribute to the performance of the classification models. This implies that, for the classification task at hand, the opponent's information may not significantly influence the prediction of player positions or categories. Among the different models utilized, support vector machines (SVM) showed superior performance specifically for the guard position in both regression and classification tasks. This implies that SVMs effectively captured the workload-efficiency relationship for guards. In addition, KNN demonstrated good performance for both the forward and center positions, with the best results observed in the classification task. This suggests that the KNN algorithm effectively captured the workload patterns and efficiency trends for these positions.

B. REGRESSION AND CLASSIFICATION RESULTS AFTER USING TVAE

1) REGRESSION

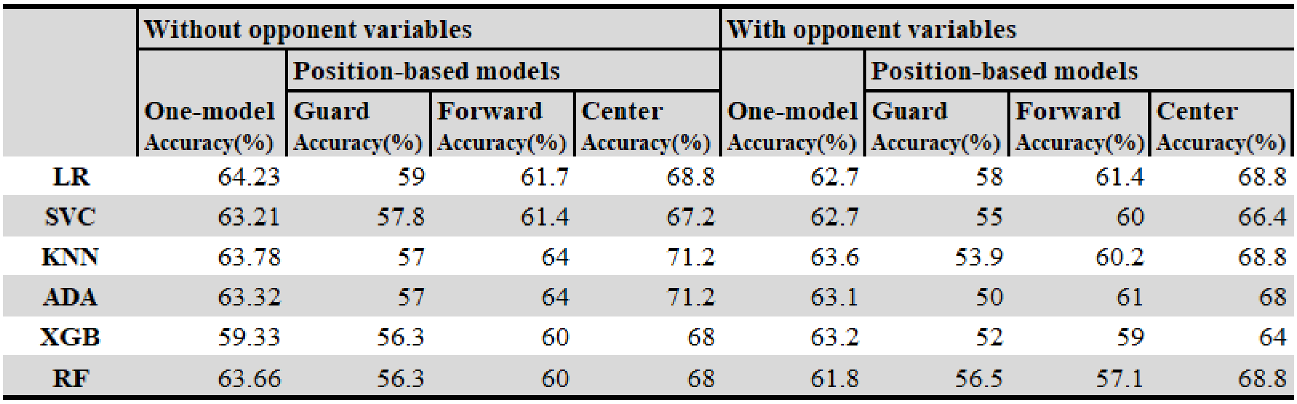
Using TVAE, 1000 and 10000 synthetic data points were generated and combined with actual data, respectively. Two sets of one-model based, and position-based regression models were built on the combined data to predict PER with and without the opponent variables included in the predictors. The regression model's performance on 1000 and 10000 data points is shown in tables 3 and 4. When the performance of these is compared with that of original data in table 1, it is observed that compared to 1000 adding 10000 data samples improved the performance of all the models. Similar performance of higher R^2 is observed with opponent variables included compared to without the opponent variables. Support vector regressor consistently performed better with 10000 samples added with higher R^2 score (0.39, 0.48, 0.29, and 0.42 for one-model, guard, forward and center positions respectively) for opponent and (0.20, 0.43, 0.28 and 0.37 for one-model, guard, forward and center respectively) for non-opponent models.

It can be observed that among all the regression models (one-model and position-based) on 10000 data points, guard position models showed higher performance followed by center and one-model approach. In all the comparisons, forward position models performed relatively low with R^2 of 0.28 and 0.29 for non-opponent and opponent models with SVR as seen in figure 2. With 1000 samples, KNN, and kernel KNN returned R^2 which is comparatively high than the other algorithms applied on these data points for forward and center positions. Random forests returned 0.23 and 0.30 R^2 for guard position with non-opponent and opponent models.

2) CLASSIFICATION

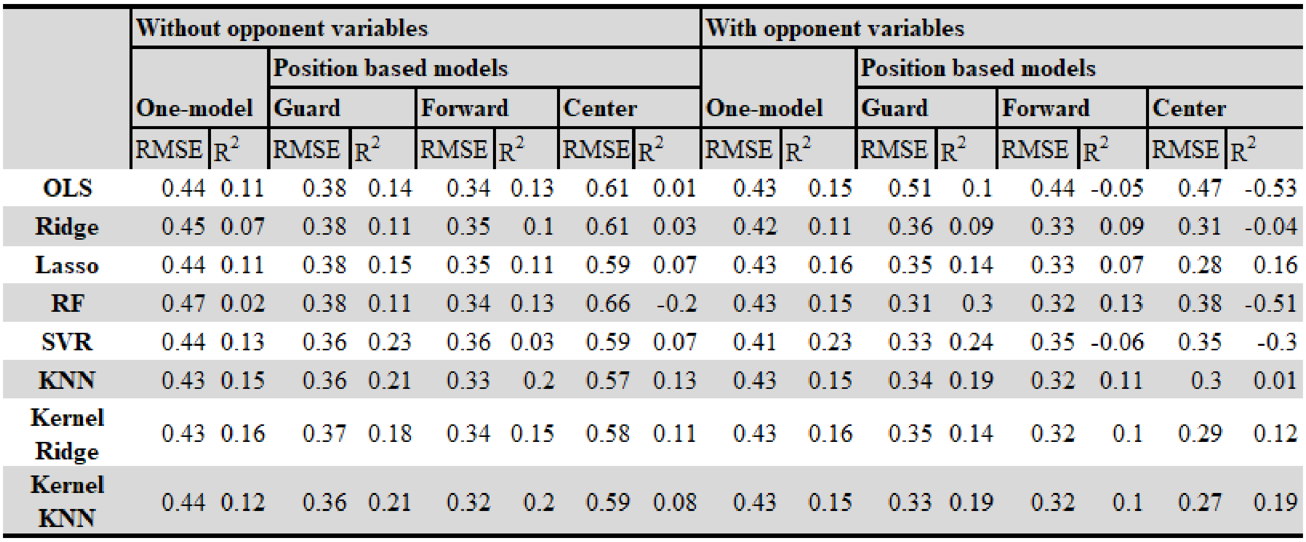
Classification models performance improved for one-model approach and for guard position in the position-based models when 10000 synthetic samples were added to the original data. 65% accuracy for one-model and nearly 68% for guard were observed with SVC without the opponent variables as seen in figure 3. Like the original data, opponent variables did not affect the classification models performance with

TABLE 2. Classification models performance.



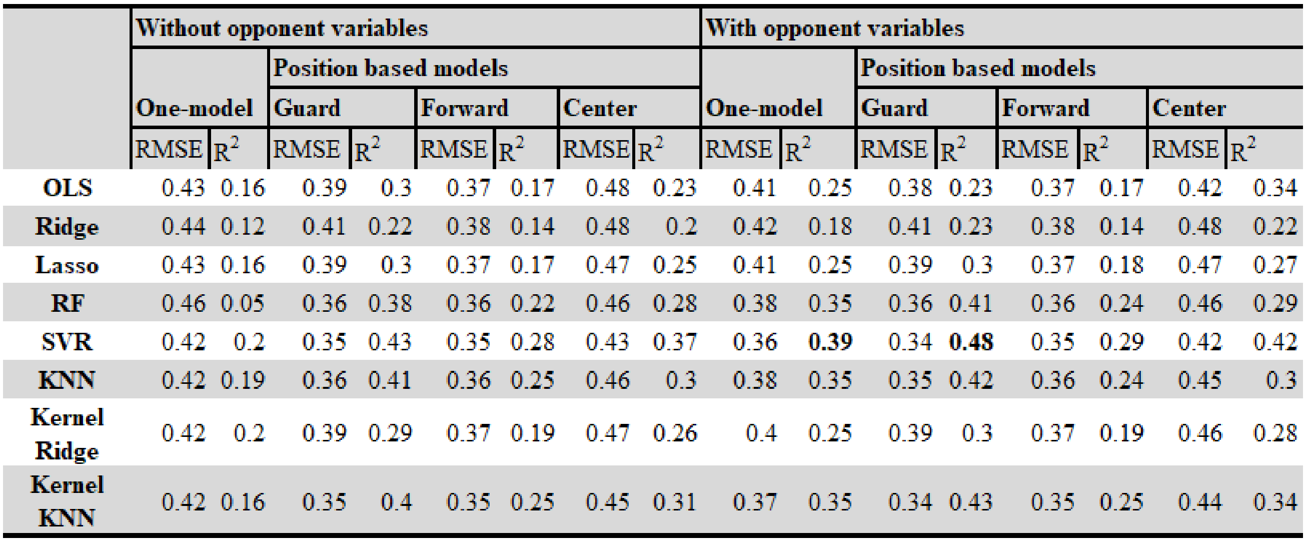
	Without opponent variables				With opponent variables			
		Position-based models				Position-based models		
	One-model Accuracy(%)	Guard Accuracy(%)	Forward Accuracy(%)	Center Accuracy(%)	One-model Accuracy(%)	Guard Accuracy(%)	Forward Accuracy(%)	Center Accuracy(%)
LR	64.23	59	61.7	68.8	62.7	58	61.4	68.8
SVC	63.21	57.8	61.4	67.2	62.7	55	60	66.4
KNN	63.78	57	64	71.2	63.6	53.9	60.2	68.8
ADA	63.32	57	64	71.2	63.1	50	61	68
XGB	59.33	56.3	60	68	63.2	52	59	64
RF	63.66	56.3	60	68	61.8	56.5	57.1	68.8

TABLE 3. Regression models performance with 1000 data points added.



	Without opponent variables								With opponent variables							
			Position based models								Position based models					
	One-model		Guard		Forward		Center		One-model		Guard		Forward		Center	
	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2
OLS	0.44	0.11	0.38	0.14	0.34	0.13	0.61	0.01	0.43	0.15	0.51	0.1	0.44	-0.05	0.47	-0.53
Ridge	0.45	0.07	0.38	0.11	0.35	0.1	0.61	0.03	0.42	0.11	0.36	0.09	0.33	0.09	0.31	-0.04
Lasso	0.44	0.11	0.38	0.15	0.35	0.11	0.59	0.07	0.43	0.16	0.35	0.14	0.33	0.07	0.28	0.16
RF	0.47	0.02	0.38	0.11	0.34	0.13	0.66	-0.2	0.43	0.15	0.31	0.3	0.32	0.13	0.38	-0.51
SVR	0.44	0.13	0.36	0.23	0.36	0.03	0.59	0.07	0.41	0.23	0.33	0.24	0.35	-0.06	0.35	-0.3
KNN	0.43	0.15	0.36	0.21	0.33	0.2	0.57	0.13	0.43	0.15	0.34	0.19	0.32	0.11	0.3	0.01
Kernel Ridge	0.43	0.16	0.37	0.18	0.34	0.15	0.58	0.11	0.43	0.16	0.35	0.14	0.32	0.1	0.29	0.12
Kernel KNN	0.44	0.12	0.36	0.21	0.32	0.2	0.59	0.08	0.43	0.15	0.33	0.19	0.32	0.1	0.27	0.19

TABLE 4. Regression models performance with 10000 data points added.



	Without opponent variables								With opponent variables							
			Position based models								Position based models					
	One-model		Guard		Forward		Center		One-model		Guard		Forward		Center	
	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2
OLS	0.43	0.16	0.39	0.3	0.37	0.17	0.48	0.23	0.41	0.25	0.38	0.23	0.37	0.17	0.42	0.34
Ridge	0.44	0.12	0.41	0.22	0.38	0.14	0.48	0.2	0.42	0.18	0.41	0.23	0.38	0.14	0.48	0.22
Lasso	0.43	0.16	0.39	0.3	0.37	0.17	0.47	0.25	0.41	0.25	0.39	0.3	0.37	0.18	0.47	0.27
RF	0.46	0.05	0.36	0.38	0.36	0.22	0.46	0.28	0.38	0.35	0.36	0.41	0.36	0.24	0.46	0.29
SVR	0.42	0.2	0.35	0.43	0.35	0.28	0.43	0.37	0.36	**0.39**	0.34	**0.48**	0.35	0.29	0.42	0.42
KNN	0.42	0.19	0.36	0.41	0.36	0.25	0.46	0.3	0.38	0.35	0.35	0.42	0.36	0.24	0.45	0.3
Kernel Ridge	0.42	0.2	0.39	0.29	0.37	0.19	0.47	0.26	0.4	0.25	0.39	0.3	0.37	0.19	0.46	0.28
Kernel KNN	0.42	0.16	0.35	0.4	0.35	0.25	0.45	0.31	0.37	0.35	0.34	0.43	0.35	0.25	0.44	0.34

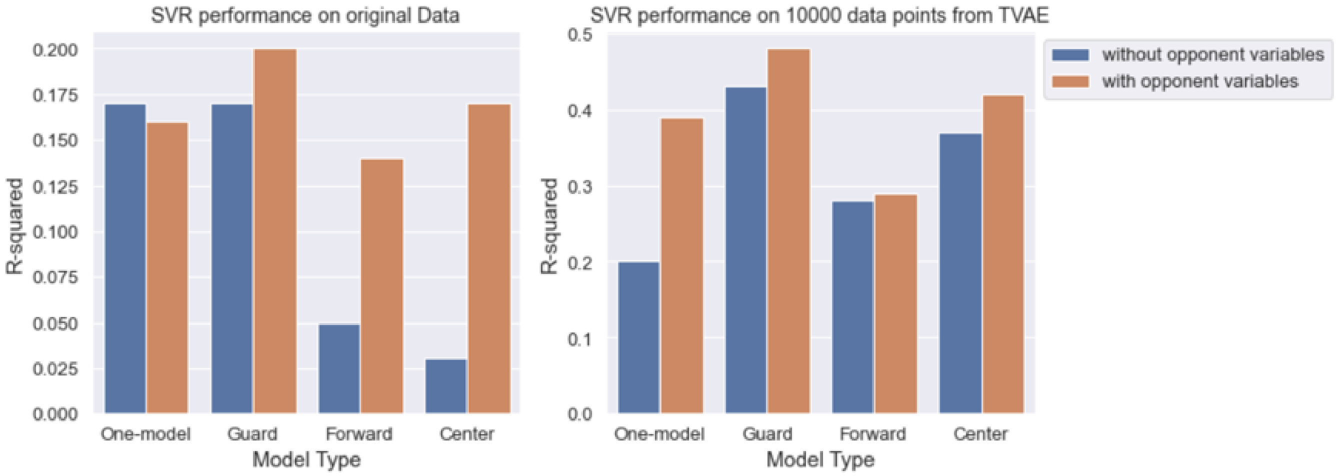


FIGURE 2. Comparison of regression models' performance.

TABLE 5. Classification models performance with 1000 data points added.

	Without opponent variables				With opponent variables			
		Position based models				Position based models		
	One-model Accuracy(%)	Guard Accuracy(%)	Forward Accuracy(%)	Center Accuracy(%)	One-model Accuracy(%)	Guard Accuracy(%)	Forward Accuracy(%)	Center Accuracy(%)
LR	60.78	56.97	59.89	69.23	61.02	56.5	57.9	67.69
SVC	56.97	66.15	61.71	61.79	59.06	56.73	61.26	63.85
KNN	55.42	52.01	61.54	69.23	62.33	54.13	58.24	65.38
ADA	62.12	53.66	64.01	48.46	61.35	55.31	51.37	48.46
XGB	58.46	54.61	55.49	60.77	62.33	55.79	54.94	69.23
RF	59.89	55.32	57.69	63.85	62.01	55.55	56.31	67.69

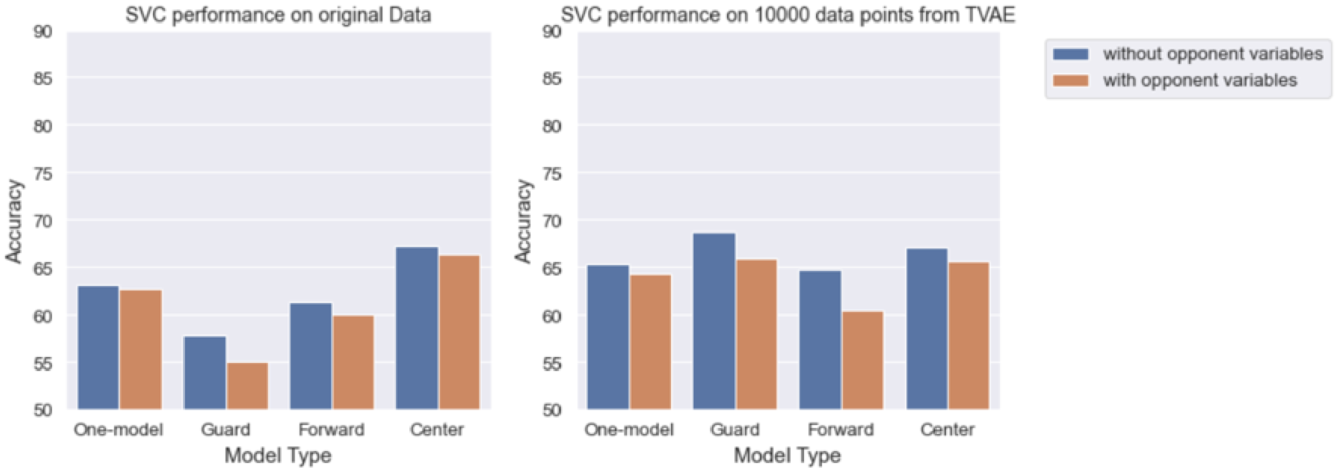
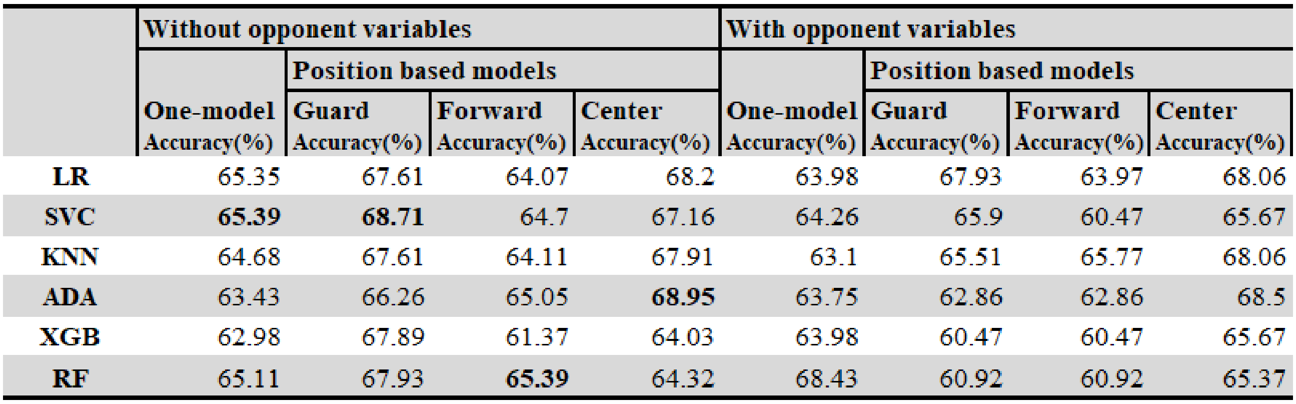


FIGURE 3. Comparison of classification models' performance.

synthetic data added. Forward and center position models appeared to have not been improved considerably by adding synthetic data. These position models had nearly 65% and 69% accuracy with both opponent and non-opponent models. The performance results of classification models with the combined data are shown in tables 5 and 6.

TABLE 6. Classification models performance with 10000 data points added.



	Without opponent variables				With opponent variables			
		Position based models				Position based models		
	One-model Accuracy(%)	Guard Accuracy(%)	Forward Accuracy(%)	Center Accuracy(%)	One-model Accuracy(%)	Guard Accuracy(%)	Forward Accuracy(%)	Center Accuracy(%)
LR	65.35	67.61	64.07	68.2	63.98	67.93	63.97	68.06
SVC	**65.39**	**68.71**	64.7	67.16	64.26	65.9	60.47	65.67
KNN	64.68	67.61	64.11	67.91	63.1	65.51	65.77	68.06
ADA	63.43	66.26	65.05	**68.95**	63.75	62.86	62.86	68.5
XGB	62.98	67.89	61.37	64.03	63.98	60.47	60.47	65.67
RF	65.11	67.93	**65.39**	64.32	68.43	60.92	60.92	65.37

VII. CONCLUSION

The results of the present study demonstrate that the workload of players, measured using catapult devices, can be effectively utilized to predict player efficiency. By incorporating a generative model, such as the Variational Autoencoder (VAE), the performance of all models was enhanced by adding 10,000 synthesized samples to the original data. This suggests that the VAE-generated data contributed to improving the predictive capabilities of the models. The study's findings indicate that independent position-based models outperformed the one-model-for-all approach in both regression and classification tasks. This highlights the importance of considering the unique workload patterns and characteristics of each player position for more accurate predictions of efficiency. Furthermore, the inclusion of opponent information had a positive impact on regression models, improving their performance. However, it had no significant effect on the classification models, implying that opponent information may not be crucial for predicting player positions or categories. The proposed algorithms achieved higher classification accuracy specifically for the center position compared to forward and guard positions. This suggests that the models were more successful in accurately classifying players into the center position compared to other positions. Additionally, the study concluded that the relationship between workload and efficiency of players is more nonlinear in the guard and forward positions. This implies that the workload's impact on player efficiency in these positions may exhibit complex and nonlinear patterns, requiring careful modeling to capture accurately. Overall, these findings contribute to the understanding of the relationship between workload and efficiency in basketball players and emphasize the importance of considering position-specific models, the inclusion of opponent information, and the nonlinear nature of this relationship in different player positions.

ACKNOWLEDGMENT

The authors would like to thank Michael Haischer, Todd Smith, and Dr. Kristof Kipp for their valuable insights and feedback and Matt Moravec for helping with synthetic data.

An earlier version of this paper was presented in part at the IEEE DSS Conference, Haikou, China, December 2021 [DOI: 10.1109/HPCC-DSS-SmartCity-DependSys53884.2021.00197].

REFERENCES

[1] T. J. Gabbett, "The training—Injury prevention paradox: Should athletes be training smarter and harder?" *Brit. J. Sports Med.*, vol. 50, no. 5, pp. 273–280, 2016.

[2] T. Reilly, T. Morris, and G. Whyte, "The specificity of training prescription and physiological assessment: A review," *J. Sports Sci.*, vol. 27, no. 6, pp. 575–589, Apr. 2009.

[3] L. B. Ransdell, T. Murray, Y. Gao, P. Jones, and D. Bycura, "A 4-year profile of game demands in elite women's division I college basketball," *J. Strength Conditioning Res.*, vol. 34, no. 3, pp. 632–638, Mar. 2020.

[4] L. Svilar and I. Jukić, "Load monitoring system in top-level basketball team: Relationship between external and internal training load," *Kinesiology*, vol. 50, no. 1, pp. 25–33, 2018.

[5] V. Manzi, S. D'Ottavio, F. M. Impellizzeri, A. Chaouachi, K. Chamari, and C. Castagna, "Profile of weekly training load in elite male professional basketball players," *J. Strength Conditioning Res.*, vol. 24, no. 5, pp. 1399–1406, May 2010.

[6] J. Vázquez-Guerrero, M. Casals, J. Corral-López, and J. Sampaio, "Higher training workloads do not correspond to the best performances of elite basketball players," *Res. Sports Med.*, vol. 28, no. 4, pp. 540–552, Oct. 2020.

[7] M. R. Román, J. García-Rubio, S. Feu, and S. J. Ibáñez, "Training and competition load monitoring and analysis of women's amateur basketball by playing position: Approach study," *Frontiers Psychol.*, vol. 9, p. 2689, 2019.

[8] J. D. Bartlett, F. O'Connor, N. Pitchford, L. Torres-Ronda, and S. J. Robertson, "Relationships between internal and external training load in team-sport athletes: Evidence for an individualized approach," *Int. J. Sports Physiol. Perform.*, vol. 12, no. 2, pp. 230–234, Feb. 2017.

[9] J. O. C. Coyne, A. J. Coutts, R. U. Newton, and G. Gregory Haff, "Relationships between different internal and external training load variables and elite international women's basketball performance," *Int. J. Sports Physiol. Perform.*, vol. 16, no. 6, pp. 871–880, Jun. 2021.

[10] J. Sampaio, E. J. Drinkwater, and N. M. Leite, "Effects of season period, team quality, and playing time on basketball players' game-related statistics," *Eur. J. Sport Sci.*, vol. 10, no. 2, pp. 141–149, Mar. 2010.

[11] J. Sampaio, M. Janeira, S. Ibáñez, and A. Lorenzo, "Discriminant analysis of game-related statistics between basketball guards, forwards and centres in three professional leagues," *Eur. J. Sport Sci.*, vol. 6, no. 3, pp. 173–178, Sep. 2006.

[12] I. Stancin and A. Jovic, "Analyzing the influence of player tracking statistics on winning basketball teams," in *Proc. 41st Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, May 2018, pp. 1533–1538.

[13] P. Beshai, "Buckets: Basketball shot visualization," Dept. Comput. Sci., Univ. Brit. Columbia, Vancouver, BC, Canada, Tech. Rep., Dec. 2014, pp. 514–547.

[14] R. Ma, D. Yan, H. Peng, T. Yang, X. Sha, Y. Zhao, and L. Liu, "Basketball movements recognition using a wrist wearable inertial measurement unit," in *Proc. IEEE 1st Int. Conf. Micro/Nano Sensors AI, Healthcare, Robot. (NSENS)*, Dec. 2018, pp. 73–76.

[15] A. Franks, A. Miller, L. Bornn, and K. Goldsberry, "Counterpoints: Advanced defensive metrics for NBA basketball," in *Proc. 9th Annu. MIT Sloan Sports Anal. Conf.*, Boston, MA, USA, Feb. 2015, pp. 1–8.

[16] N. Kuhlman and C.-H. Min, "Analysis and classification of basketball shooting form using wearable sensor systems," in *Proc. IEEE 11th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2021, pp. 1478–1482.

[17] M. Sadeghi and X. Alameda-Pineda, "Mixture of inference networks for VAE-based audio-visual speech enhancement," *IEEE Trans. Signal Process.*, vol. 69, pp. 1899–1909, 2021.

[18] T. J. Vandal, D. McDuff, W. Wang, K. Duffy, A. Michaelis, and R. R. Nemani, "Spectral synthesis for geostationary satellite-to-satellite translation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4702611.

[19] J. Bae and C. Lee, "Easy data augmentation for improved malware detection: A comparative study," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Jan. 2021, pp. 214–218.

[20] D. Barrejón, P. M. Olmos, and A. Artés-Rodríguez, "Medical data wrangling with sequential variational autoencoders," *IEEE J. Biomed. Health Informat.*, vol. 26, no. 6, pp. 2737–2745, Jun. 2022.

[21] C. Zhang and Y. Peng, "Stacking VAE and GAN for context-aware text-to-image generation," in *Proc. IEEE 4th Int. Conf. Multimedia Big Data (BigMM)*, Sep. 2018, pp. 1–5.

[22] J. Liang and J. Chen, "Data augmentation of thyroid ultrasound images using generative adversarial network," in *Proc. IEEE Int. Ultrason. Symp. (IUS)*, Sep. 2021, pp. 1–4.

[23] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional GaN," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.

[24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–9.

[25] J. L. Devore, *Probability and Statistics for Engineering and the Sciences*. Boston, MA, USA: Cengage Learning, 2011.

[26] A. Kolmogorov, "Sulla determinazione empirica di una lgge di distribuzione," *Inst. Ital. Attuari, Giorn.*, vol. 4, pp. 83–91, 1933.

[27] D. Lopez-Paz and M. Oquab, "Revisiting classifier two-sample tests," 2016, *arXiv:1610.06545*.

[28] S. Bourou, A. El Saer, T.-H. Velivassaki, A. Voulkidis, and T. Zahariadis, "A review of tabular data synthesis using GANs on an IDS dataset," *Information*, vol. 12, no. 9, p. 375, Sep. 2021.

[29] A. S. Gwelo, "Principal components to overcome multicollinearity problem," *Oradea J. Bus. Econ.*, vol. 4, no. 1, pp. 79–91, Mar. 2019.

[30] C. M. Bishop and N. M. Nasrabadi, *Pattern Recognition and Machine Learning*, vol. 4, no. 4. New York, NY, USA: Springer, 2006, p. 738.

[31] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.

[32] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.

[33] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proc. 31st Int. Conf. Mach. Learn.*, Jun. 2014, pp. 1278–1286.

[34] N. Patki, R. Wedge, and K. Veeramachaneni, "The synthetic data vault," in *Proc. IEEE Int. Conf. Data Sci. Adv. Analytics (DSAA)*, Oct. 2016, pp. 399–410.

NASIM YAHYASOLTANI (Member, IEEE) received the B.Sc. degree in electrical and computer engineering from the University of Tehran, Tehran, Iran, in 2003, the M.Sc. degree in electrical engineering from the Iran University of Science and Technology, Tehran, in 2006, and the Ph.D. degree in electrical engineering from the University of Minnesota, Twin Cities, in June 2014. She was a Research Associate with the Digital Technology Center, University of Minnesota, from 2014 to 2017. She was a Research Associate with the University of Wisconsin–Milwaukee, from 2017 to 2018. From 2018 to 2019, she was a Senior Data Scientist with Harley-Davidson Motor Company. Since August 2019, she has been a Northwestern Mutual Assistant Professor with the Department of Computer Science, Marquette University. Her research interests include statistical signal processing, machine learning, optimization theory, and network science with applications to wireless communications and networking, big data analytics, healthcare, and smart grid.

PRIYANKA ANNAPUREDDY received the B.E. degree in electrical and electronics engineering from Sri Krishnadevaraya University, Anantapur, India, in 2006, the M.E. degree in power electronics from Jawaharlal Nehru Technological University Hyderabad, Hyderabad, India, in 2008, and the M.S. and Ph.D. degrees in computer science from Marquette University, Milwaukee, WI, USA, in 2022. Her research interests include artificial intelligence, machine learning, interpretability, time series analysis, data analytics, crisis theory, and mental health.

MANZUR FARAZI received the B.S. degree in statistics from Jahangirnagar University, Bangladesh, the M.S. degree in statistics from Ball State University, Muncie, IN, USA, the M.B.A. degree from the University of Dhaka, Bangladesh, and the Ph.D. degree in computational science from Marquette University, Milwaukee, WI, USA, with a focus on concentration in statistics and data science. After completing the Ph.D. degree, he joined the Medical College of Wisconsin as a Biostatistician with the Department of Pediatric Surgery. He is a Statistical Research Scientist II with the Division of Biostatistics, Medical College of Wisconsin. He is currently a Highly Accomplished Data Scientist and a Biostatistician with extensive experience in both academia and industry. His research interests include data science and statistical genomics. He is also an Active Member of several professional societies, including the American Statistical Association and the Institute of Mathematical Statistics.