

A Game of Runs: Extracting Initiation Rules for Basketball Scoring Runs through Focused Sequence Mining

Ioannis Sevrissarianos and Ioannis Katakis

Department of Computer Science,

School of Sciences and Engineering,

University of Nicosia, 2417, Nicosia, Cyprus

Abstract

Basketball games are often defined by scoring runs, where one team gains momentum while the opponent struggles to respond. Although it is widely recognized that such runs frequently decide the outcome of games, predicting when they will occur remains unexplored. This research addresses three key areas. First, we develop an algorithm to identify patterns that lead to scoring runs. Second, we present a model capable of predicting the onset of these runs. Finally, building on these findings, we propose a generative model that, based on partial sequences, suggests full sequences likely to result in a run. Our predictor employs a novel pattern recognition method using convolutional neural networks, achieving an accuracy of 81%. To our knowledge, this is the first study to explore and predict basketball scoring runs using this approach.

KEYWORDS: BASKETBALL, SCORING RUNS, SEQUENCE MINING, PLAY-BY-PLAY, CNNs

Introduction

Quantitative data is increasingly being gathered for multiple team sports over the past decade. That data usually covers the span of a game, an entire season or play-by-play events. Analyses conducted on gathered data are often vital in guiding coaching staff decisions, assisting players in improving their skills (Severini, 2020), or even settling front office dilemmas regarding contracts and trades (Maymin, 2017). Basketball is one sport where the combination of its complex nature along with the density and repeatability of in-game events make it an ideal candidate for exploring its play-by-play elements. Basketball is, also, often the venue where a phenomenon known as a 'scoring run' is encountered. Scoring runs are defined as periods where one team scores consecutively and builds momentum. Such runs can often play a critical role in the momentum of a game, and by extension on its outcome (Morgulev et al., 2020). Justifiably, basketball is frequently then referred to as 'a game of runs' by pundits.

For this study, the goal is to try and learn from play-by-play data. Traditionally, box scores have been used to train models to predict game outcomes (Loeffelholz et al., 2009). However, this study shifts focus to a finer temporal scale, aiming to predict scoring runs based on play-by-play data. By analyzing in-game sequences, we can identify patterns leading to momentum shifts and leverage this knowledge for strategic in-game decision-making. To achieve this, we propose an innovative approach utilizing focused sequence mining and deep learning techniques.

One of the core challenges in analyzing scoring runs is identifying specific event sequences that precede their initiation. This study approaches the problem through focused sequence mining, which involves extracting meaningful, high-frequency event patterns that contribute to scoring runs. An example of such an event pattern can be seen in Fig. 1.

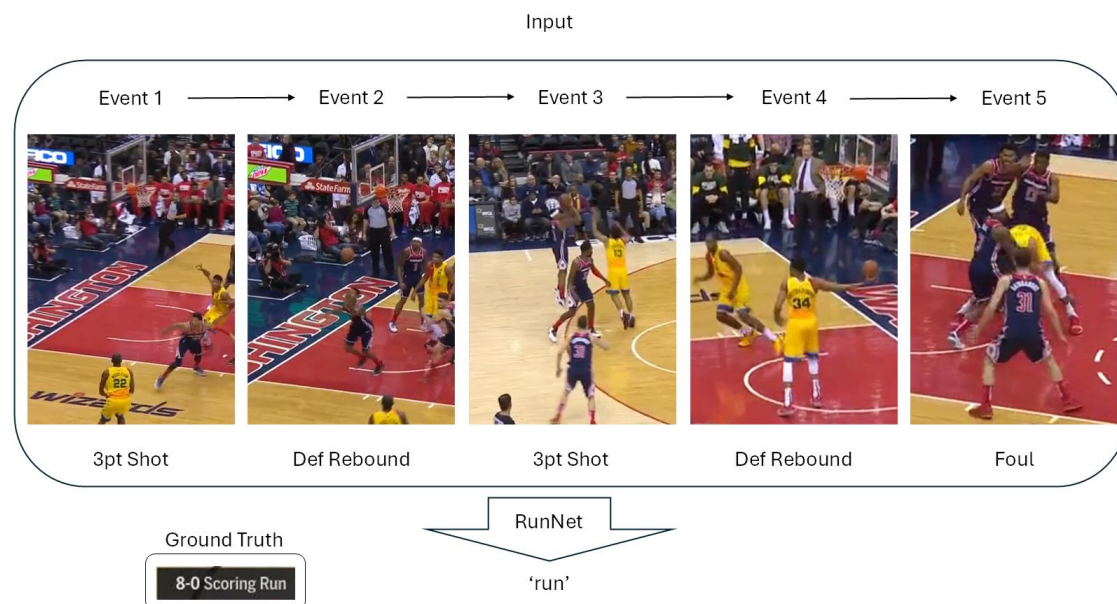


Fig 1. A sequence of five events leading to a scoring run

Unlike traditional sequence mining, which detects general trends in large datasets, focused sequence mining is tailored to capture the nuances of specific outcomes—in this case, the initiation of scoring runs. By applying this technique, we can systematically determine which sequences of game actions, such as shot attempts, defensive fouls, and rebounds, play a decisive role in triggering momentum shifts.

Our methodology leverages a structured framework that isolates these sequences and distinguishes between home and away team runs. By extracting common patterns from historical data, we aim to identify actionable insights that can help teams anticipate and react to emerging momentum shifts. This contributes to a broader understanding of game dynamics and provides a basis for predictive modeling.

Moreover, we develop a convolutional neural network (CNN) architecture, dubbed ‘RunNet’, to learn from events precipitating a scoring run. CNNs are a class of deep neural networks particularly well-suited for pattern recognition tasks due to their ability to capture spatial hierarchies in data. Originally designed for image processing, CNNs have been successfully applied to various domains, including natural language processing (Yin et al., 2017) and time series analysis (Livieris et al., 2020). The application of CNNs to predict scoring runs in basketball represents a novel foray into sports analytics, extending deep learning into the realm of fine-grained, play-by-play event data. A review of the literature, as seen in Section 2, confirms a gap, as prior research has leveraged mostly reinforcement learning or recurrent neural networks (RNNs) for predicting broader outcomes. However, none of these approaches were designed to predict the onset of scoring runs in a supervised learning context. To our knowledge, no prior study has employed CNNs to analyze sequences of play-by-play events with the specific aim of forecasting a scoring run. This aligns with recent observations that the use of convolutional networks for pattern recognition in in-game event sequences remains largely unexplored, apart from applications to player movement or positioning data.

The input data for our CNN consists of a series of play-by-play events, each represented as a vector of features. These features include details such as the type of shot, distance, player actions, and game context. Each event in a sequence is encoded into a high-dimensional vector, forming a matrix where each row represents an event, and each column corresponds to a specific feature. This matrix is analogous to an image in traditional CNN applications, where each pixel’s intensity is replaced by the numerical value of a feature.

By treating sequences of play-by-play events as structured data, the CNN can learn to recognize the complex patterns that lead to scoring runs. This approach leverages spatial and temporal dependencies in the data, capturing how specific sequences of actions and game states contribute to shifts in momentum.

Additionally, we introduce a generative algorithm, based on the trained RunNet model, designed to complete partial sequences in real time. Thus, optimizing event sequences to maximize the probability of a scoring run. This model is particularly valuable in strategic decision-making, allowing coaches and analysts to simulate potential game scenarios and adjust strategies accordingly. By treating the problem as a sequence generation task, we provide a novel approach that extends beyond traditional game predictions to real-time decision support.

Contribution. This paper’s contribution can be summarized in the following points:

1. We frame the problem of predicting scoring runs in basketball as a sequence mining problem.
2. We propose a sequence mining algorithm to identify the most frequent event patterns that lead to scoring runs, distinguishing between home and away team runs. The algorithm is specifically designed for the present problem. Additionally, we are exploring these sequences on the team and season level, allowing for more specific sequences to emerge. To our knowledge this is the first attempt at extracting such patterns.

3. We introduce an approach utilizing a convolutional neural network (CNN) model capable of identifying and predicting the initiation of scoring runs in NBA basketball games using play-by-play data. The model achieves 81% accuracy in predicting the onset of scoring runs, outperforming traditional classifiers.
4. We introduce a generative model that completes partial sequences of play-by-play events, predicting the likelihood of a scoring run based on incomplete data. This becomes increasingly useful in in-game and executive environments.
5. We develop an API that allows a user to leverage the researched approaches for their own datasets.

Related Work

The current literature, so far, has not been greatly involved with making predictions on a temporal scale of the next few game events. The often-favourable aspect of predicting entire game or season outcomes for betting purposes has until now been a significant motivator for a wider temporal resolution. At the same time, coaching staff, most frequently, make data-driven changes between, rather than during games. However, with the advent of improved hardware and computational efficiency, such data can be analyzed on the court and provide insights for on-the-fly decision making. At the same time, the use of convolutional networks for pattern recognition during game events has been largely unexplored, bar use for movements and positioning. Below, we outline various approaches used in the context of basketball games, while highlighting areas other than image processing where convolutional networks have proven successful in the past.

Sequence Mining for Sports and Event Data

Sequence mining, the task of identifying recurrent patterns within ordered data, has been extensively studied across domains such as bioinformatics, retail transactions, and web activity logs. Foundational algorithms like AprioriAll (Agrawal & Srikant, 1995), PrefixSpan (Han et al., 2001), and SPADE (Zaki, 2001) established efficient frameworks for discovering frequent subsequences in large databases. These approaches have been successfully extended to mine temporal dependencies (Mannila et al., 1997) and incorporate contextual constraints such as time gaps or event hierarchies. However, while these algorithms are powerful for general-purpose sequence discovery, they typically focus on frequency or support thresholds, without explicitly considering outcome-driven or label-dependent relationships. This is a limitation when the goal is to identify sequences predictive of a particular result, such as scoring runs.

In sports analytics, sequence mining has been applied only sparingly and often at a coarse level of temporal abstraction. Vračar et al. (Vračar et al., 2016) utilized sequence mining over basketball play-by-play data to model event transitions, but the objective was descriptive rather than predictive. Sarlis et al. (Sarlis et al., 2024) and Weimer et al. (Weimer et al., 2023) analyzed temporal relations in injury and momentum data, respectively, though their methods relied on association rules or causal inference rather than explicit sequential pattern extraction. In soccer, Perin et al. (Perin et al., 2018) and Decroos et al. (Decroos et al., 2019) employed pattern mining to identify recurring tactical plays, but these approaches emphasized high-level possession structures rather than granular, labeled event sequences. Beyond sports, specialized adaptations of sequential pattern mining, such as contrast mining (Ji et al., 2014) and discriminative sequence mining (Cheng et al., 2008), have been proposed to distinguish between classes of sequences, yet these algorithms typically assume fixed-length patterns or rely on pre-defined features, limiting their ability to generalize to complex, heterogeneous event streams like basketball play-by-play data.

Convolutional Networks in Non-image-related Tasks

Despite finding significant success in image processing tasks, CNNs have often been utilized in tasks of other natures as well. (Karthikeyan & Priyadharsini, 2024) have used CNNs in combination with attention mechanisms in an effort to recognize voiceprints invariant of text values. In (Sun et al., 2024) high-speed electrical multiple units (EMUs) defects are being detected using a convolutional network architecture. Drainage networks are being studied in (Xu et al., 2023), where CNNs prove invaluable in mapping the intricacies and complexities of river patterns. Emotion recognition based on electro-encephalogram features is being accommodated by cascaded CNN layers in (Meng et al., 2023).

Several factors make CNNs particularly attractive for the present study and their utilization in non-image-related tasks. First, CNNs are adept at uncovering latent temporal patterns in sequences of game events. By sliding across the play-by-play sequence, a convolutional network can learn which local configurations of events signal an impending run. This might include subtle cues such as a cluster of missed shots and defensive rebounds or a rapid succession of turnovers.

Second, CNN models offer the benefit of efficient, real-time inference, which is crucial for in-game applications. Once trained, a CNN can ingest new sequences of plays and output a prediction with very low latency, thanks to parallelizable convolution operations. This real-time potential means a scoring run prediction model could be deployed during live games, giving coaching staff actionable insights. Notably, as hardware and computational resources have improved, on-the-fly analysis of sports data has become feasible. Our CNN-based approach could continuously monitor the sequence of plays and issue an alert when a pattern resembling the start of a run emerges, effectively being a momentum early-warning system. The implications for coaching and in-game strategy are significant: if a model can predict an impending run even a few possessions in advance, coaches could proactively call a timeout, adjust defensive matchups, or make substitutions to preempt or mitigate the run. This kind of data-driven, moment-to-moment decision support has not been practical historically, as most analytics were done post-game or at halftime.

Third, applying CNNs in this context pushes sports analytics into a new regime of granularity and insight. Earlier studies on momentum or “hot hands” in basketball often relied on coarse metrics or simplified models, and they sometimes debated whether runs are statistically significant or just random variance. A deep convolutional model, in contrast, can provide a nuanced perspective by highlighting which specific sequences of micro-actions contribute to a genuine momentum shift.

Momentum in Sports

The concept of momentum, the perceived psychological or performance-based shift favoring one team or player, has been a central yet contentious topic in sports science and analytics. Early psychological theories (Iso-Ahola & Dotson, 2014) described momentum as a self-reinforcing cycle in which successful outcomes increase confidence and subsequent performance. However, empirical studies have produced mixed evidence regarding its existence or measurability. From an analytical perspective, momentum is often operationalized as sequences of successful plays or scoring bursts, commonly referred to as runs in basketball and other sports.

In basketball, numerous studies have attempted to quantify momentum using aggregate statistics or probabilistic models. Morgulev et al. (Morgulev et al., 2020) examined overtime data and found that “comeback” teams did not exhibit significantly higher win probabilities, suggesting that the perception of momentum may exceed its statistical manifestation.

Conversely, Weimer et al. (Weimer et al., 2023) employed a causal inference framework to detect team-level momentum, reporting that short-term scoring surges could be identified statistically and that their persistence depended on contextual factors such as timeouts and game phase. Gibbs et al. (Gibbs et al., 2022) investigated the efficacy of timeouts in halting opponents' runs, concluding that timeouts have limited and short-lived impact on reversing momentum. Similarly, Tenenbaum et al. (Tenenbaum et al., 2024) analyzed two decades of NBA data and observed that technical fouls can act as momentum triggers, especially for home teams. These works collectively reinforce the notion that scoring runs serve as observable correlates of momentum shifts, even if their psychological underpinnings remain elusive.

Beyond basketball, momentum-related phenomena have been explored in tennis, soccer, and hockey. (Klaassen & Magnus, 2014) modeled tennis momentum as conditional probabilities of point wins following streaks and found evidence of “mini runs” consistent with behavioral inertia. In soccer, (Ötting et al., 2023) employed a copula-based multivariate hidden Markov model to identify phases of increased scoring likelihood, framing them as manifestations of tactical momentum. In ice hockey, (Ryall & Edgar, 2022) employed sequence-based modeling to detect shifts in team dominance across game segments. These studies demonstrate a growing interest in quantifying dynamic performance fluctuations across sports through event-level temporal data.

Despite these advances, most prior research treats momentum descriptively, identifying its occurrence retrospectively rather than predicting its onset. Moreover, statistical and causal models typically depend on aggregated or smoothed data rather than fine-grained event sequences. As such, the mechanisms that initiate momentum remain underexplored. The present study addresses this gap by reframing the notion of momentum in basketball as a predictive sequence classification problem, where scoring runs are used as quantifiable manifestations of momentum

Settings for broader predictions

Research on predictive modeling in basketball and related invasion sports has moved from primarily forecasting final outcomes toward a spectrum of targets spanning game results, in-game state/value estimates, and season-long performance trajectories. Broad reviews of sport outcome prediction characterize most match forecasting as a supervised classification problem whose reported success is strongly shaped by feature construction and, crucially, by evaluation design that respects temporal ordering to avoid leakage (Horvat & Job, 2020; Bunker & Susnjak, 2022; Bunker & Thabtah, 2019). In basketball specifically, a recent systematic review of AI-based outcome prediction in professional leagues (2019–2024) documents heavy use of neural and ensemble approaches and emphasizes the recurring role of data processing, feature selection, and increasingly dynamic prediction settings intended for tactical or decision-support use (Li et al., 2025). Complementing “who wins” forecasts, contemporary work in other invasion sports illustrates how ML pipelines can target tactically meaningful intermediate objectives, such as predicting whether possession will reach a high-threat region (Stival et al., 2023) or estimating possession/action value from spatiotemporal context via expected possession value and related action-valuation frameworks (Fernández et al., 2021; Xarles et al., 2025). Finally, reviews of team-sport AI applications show that longer-horizon prediction problems (performance development, injury risk, and talent identification) are increasingly tackled with ML but often face limitations in data heterogeneity and external validation that constrain real-world generalizability (Claudino et al., 2019; Tang et al., 2026).

Problem Definition

In this study, we define a **scoring run** as a period of increased momentum for the team holding it. That period starts with one team scoring a basket (free throw, 2-point shot, 3-point shot) and ending with the opposing team scoring any basket. Our definition requires that period to consist of a minimum of three made baskets. Crucially, runs are treated as maximal, non-overlapping events; a continuous streak of scoring is recorded as a single run regardless of its total length, rather than being segmented into multiple smaller sub-sequences. We analyze three key problems related to scoring runs in game event sequences:

6. Identifying sequences of game events that lead to scoring runs.
7. Training a classifier to predict whether a sequence can lead to a scoring run.
8. Generating missing parts of sequences to maximize the probability of a scoring run.

Focused Sequence Mining

Consider a sequence of discrete game events:

$$E = (e_1, e_2, \dots, e_n),$$

where each e_i represents an event such as a shot attempt, foul, or rebound. Our goal is to extract a subsequence

$$R = (e_i, e_{i+1}, \dots, e_{i+k}),$$

where $R \subseteq E$, meaning R is a subset of E and consists of contiguous events from E and $k < n$, such that R leads to a scoring run.

Formally, let $S_{\text{team1}}(t)$ and $S_{\text{team2}}(t)$ be the scores of the two teams at time t . The scoring run condition can be written as:

$$\Delta S(t) = S_{\text{team1}}(t) - S_{\text{team2}}(t),$$

where

$$\Delta S(t) > \Delta S(t-1) \forall t \in [t_0, t_0 + T],$$

for a fixed duration T , and with $S_{\text{team2}}(t) = S_{\text{team2}}(t_0)$ remaining unchanged over this period.

Runs Prediction

The objective here is to classify whether a given event sequence leads to a scoring run. This is formulated as a supervised learning problem where we learn a function:

$$f: (E_{t-W}, \dots, E_t) \rightarrow \{\text{run}, \text{no_run}\},$$

where W is a fixed window size and f is a classifier that predicts the probability of a run given the preceding event sequence.

To train f , we define a labeled dataset where each input sequence is annotated based on whether it leads to a run. The classifier is optimized using a loss function such as binary cross-entropy:

$$L(f) = - \sum_i [y_i \log f(E_i) + (1 - y_i) \log(1 - f(E_i))],$$

where y_i is the ground-truth label (1 for run, 0 for no run).

Sequence Generation

In cases where only a partial sequence is available, we aim to generate a completion that maximizes the probability of leading to a scoring run. Given a partial sequence:

$$P = (e_i, e_{i+1}, \dots, e_{i+m}),$$

where $m < k$, we seek to generate the missing events:

$$G = (e_{i+m+1}, \dots, e_{i+k}),$$

such that the full sequence $P \cup G$ maximizes the probability of a scoring run:

$$G = \underset{G}{\operatorname{argmax}} P(\text{run} \vee P \cup G).$$

Providing solutions to the above problems can result in a better understanding of the game events that contribute to the initiation of a scoring run, while also allowing us to guide in-game actions for improved team scoring performance.

Data

In order to gain insight into the moments of a game, it was necessary to utilize play-by-play data. The data that was scraped from basketball-reference.com covers NBA seasons from 2015 to 2020. The data is formatted in such a way that each game event is indicated by a different row and includes a total of 40 features. In Table 1 we present an example of the total features of the original dataset. As seen, a subset of the features is of descriptive nature and could not contribute to meaningful predictions in our current context.

Table 1. Full Data Sample

Feature	Value	Feature	Value
URL	/boxscores/201910240GSW.html	FoulType	N/A
GameType	regular	Fouler	N/A
Location	Chase Center San Francisco California	Fouled	N/A
Date	October 24 2019	Rebounder	N/A
Time	10:30 PM	ReboundType	N/A
WinningTeam	LAC	ViolationPlayer	N/A
Quarter	1	ViolationType	N/A
SecLeft	297	TimeoutTeam	N/A
AwayTeam	LAC	FreeThrowShooter	N/A
AwayPlay	N/A	FreeThrowOutcome	N/A
AwayScore	20	FreeThrowNum	N/A
HomeTeam	GSW	EnterGame	N/A
HomePlay	S. Curry makes 2-pt jump shot from 21 ft	LeaveGame	N/A
HomeScore	14	TurnoverPlayer	N/A
Shooter	S. Curry - curryst01	TurnoverType	N/A
ShotType	2-pt jump shot	TurnoverCause	N/A
ShotOutcome	make	TurnoverCauser	N/A
ShotDist	21	JumpballAwayPlayer	N/A
Assister	N/A	JumpballHomePlayer	N/A
Blocker	N/A	JumpballPoss	N/A

Out of the 40 features, 11 were used for sequence mining, fitting classifiers and training our model, as seen in Table 2. For demonstration purposes, we group them into categories representative of their contribution. These categories do not impact sequence mining or further model training. Each row of those features describes an event. The features were engineered so that values were either binned or described at a Boolean level. Additionally, all features with the exception of 'Shot Distance' and 'Rebound Type' adhere to a scheme describing whether the action was undertaken by the team that will eventually initiate the run or their opponents. Shot distance values are binned as '3pt', 'close', 'mid', while Rebound Type contains values 'defensive' and 'offensive'. Those were decided as factors that satisfy two conditions. Firstly, they are all factors that can affect the current game state. Secondly, they are all features that can take variable values over the course of a game.

Table 2. Features

Shooting	Violations	Defense	Rebounding	Misc
Shooter	Violation Player	Blocker	Rebounder	Timeout Team
Free Throw Shooter			Rebound Type	Substitution
Shot Distance				

As described in the Methodology section, a critical step is to process the raw play-by-play data (Table 1) into structured event sequences suitable for modeling. This transformation follows a clear sequence.

- First, each individual play-by-play event is encoded using only the 11 selected features detailed in Table 2.
- Next, to capture the context leading up to a scoring run, we group these encoded events into fixed-size windows, specifically taking the 10 consecutive actions that occur immediately preceding the start of a potential scoring run (or a corresponding non-run segment).
- Finally, each group of 10 events is assigned the appropriate label (run or non-run) and reshaped into an 11×10 matrix. In this structure, one dimension represents the 11 features and the other represents the 10-event sequence. This matrix becomes the direct input for the RunNet.

This process creates the connection between the raw data in Table 1, the feature selection in Table 2, and the final model input. In Table 3 we present part of the transformed data, before the final reshaping, as an example.

Table 3. Transformed Data Example

ShotDist1	Shooter1	Rebounder1	Fouler1
...	...	...	...
NaN	NaN	NaN	fouler_away
NaN	NaN	rebounder_away	NaN
close	shooter_home	NaN	NaN
3pt	shooter_away	NaN	NaN
NaN	NaN	NaN	NaN
...	...	...	...

In Figure 2 we display the total runs detected by our algorithm per season. We can note how the 2019-20 produced significant less runs, which can be attributed to the fewer games played and the 'pandemic bubble conditions (during the 2019–2020 NBA season restart)' conditions during the latter part of the season. Additionally, it is evident how more runs tend to be recorded for the home rather than the away team.

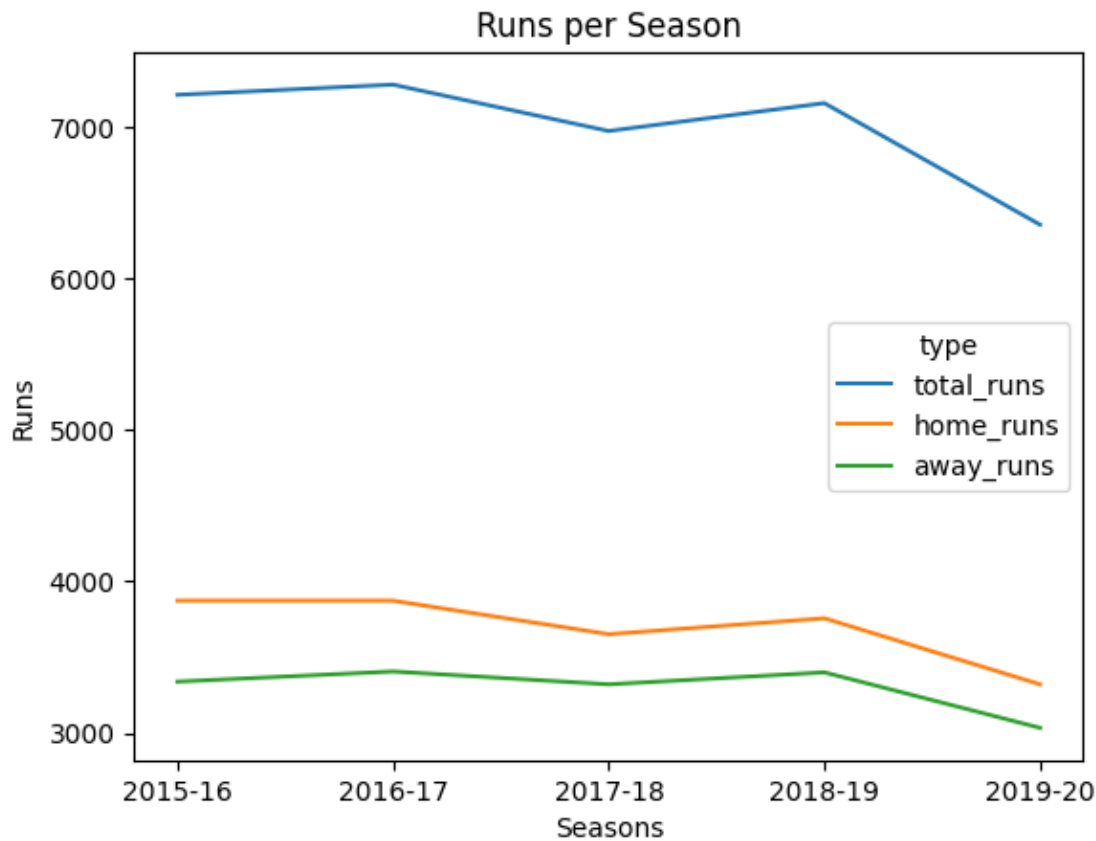


Fig 2. Runs Recorded per Season

Methodology

Focused Sequence Mining

The first goal was to identify when a run occurred. This was achieved by taking advantage of the 'HomePlay' and 'AwayPlay' features. Every time a team scored a basket, regardless of way, the word 'makes' would appear as part of the value of one of these features. By finding consecutive appearances of that word in one of the features, without it appearing in the other, we could determine that a run has occurred. For the purposes of this study, we limit the number of appearances to four. As a result, the definition of a 'run' in the current context is confined to four consecutive baskets made. By now knowing when a run starts, the next task was to define the time period of interest. Based on the research question, the ten most recent events before the start of a run were taken into consideration. Therefore, we arrive at an 11-dimension feature vector for each event. Each 11×10 matrix was converted into a single vector and assigned a label as a 'run'. Similarly, all vectors that did not precede a run were assigned a 'no_run' class. We further split the dataset into two, depending on whether the home or away team initiated the run. Our hypothesis is that there are certain sequences of actions that will lead to a scoring run.

Since we now know all sequences that can lead to a run, the next step is to find the most common ones. The process to do so can be seen in Algorithm 1, where we present in pseudocode a step-by-step representation of the workflow, designed to make the logic reproducible and transparent. The procedure **SequenceMining(df)** receives a play-by-play dataset where each row encodes a fixed-length window of events and a binary label **class** *run*, *no-run* indicating whether that window immediately precedes the start of a scoring run. The goal is to discover, for multiple pattern lengths **L**, the most frequent contiguous event pattern observed among the **run**-labeled windows, together with (i) the indices of windows that exhibit that pattern, (ii) the pattern length, and (iii) its relative frequency among all **run** windows.

The algorithm iterates over a grid of pattern configurations (the outer **for** loop over **j**, **event**). Here, **j** selects the prefix of columns corresponding to the first **L** events in each window (recall that each window is represented by a fixed ordering of event–feature slots; increasing **j** effectively increases **L**), while **event** iterates over pattern lengths from 10 down to 1. For each length **L**, we restrict the dataframe to **run** windows and take only the columns that encode the first **L** events, yielding a submatrix **X**.

Within each length **L**, the algorithm counts identical rows of **X** (each row is converted to a hashable **row_tuple**), thereby tallying how often exactly the same event pattern occurs across **run** windows. This produces a dictionary **row_counts** mapping unique patterns to their counts. The pattern with the largest count is recorded as **most_common_row**, and its support (absolute frequency) is stored as **max_count**.

To aid downstream analysis and visualization, the procedure also returns the indices of **run** windows that realize the most common pattern at length **L** **mc_indices**. In addition, it computes a relative frequency for that pattern,

$$max = max_{count} / total_{sequences}$$

which expresses how prevalent the discovered pattern is among all **run** windows of the dataset. These values are appended to the output lists: **events_idx** (indices per length), **pattern_lengths** (the explored **L**), and **frequencies** (the corresponding ratios).

Intuitively, running this procedure over decreasing lengths **L** provides a hierarchy of patterns: long, highly specific sequences (larger **L**) and shorter, more general ones (smaller **L**). Reporting both absolute counts and ratios allows us to compare patterns fairly across lengths and datasets. Note that the mining is label-aware (it only considers **class = run**) and contiguity-preserving (patterns are prefixes of the fixed window), which makes the extracted patterns directly interpretable as pre-run event chains rather than arbitrary itemsets.

Finally, the outputs, **events_idx**, **pattern_lengths**, and **frequencies**, are used to (i) populate the rules tables (e.g., most common “Last **L**” patterns for home/away runs), (ii) quantify how strongly each pattern characterizes run initiation, and (iii) retrieve representative windows for qualitative inspection and case studies.

```

Procedure: SequenceMining(df)
  Input: A dataframe 'df' with event sequences and a binary 'class' label.
  Output: events_idx, pattern_lengths, frequencies

  events_idx <- {}           // Stores indices of maximum-count events
  total_sequences <- len(df[df['class'] == 'run'])
  pattern_lengths <- []      // Stores the lengths of discovered patterns
  frequencies <- []         // Stores frequencies of the patterns

  For j, event in {range(12, 112, 11)} x {range(10, 0, -1)}:
    // Subset the dataframe by event and class
    X <- df.iloc[:, :j-1][df['class'] == 'run']

    // Dictionary to store row counts for each event sequence
    row_counts <- dict(int)

    For i in 1 to len(X):
      row_tuple <- tuple(X.iloc[i]) // Convert row i into a tuple
      row_counts[row_tuple] += 1    // Increment count for this tuple
    EndFor

    max_count <- 0                // Variable to track the maximum count
    most_common_row <- None       // Stores the most frequent event sequence

    For row, count in row_counts.items():
      If count > max_count:
        max_count <- count
        most_common_row <- row
      EndIf
    EndFor

    // Find indices of most common row
    mc_indices <- X.apply(lambda row: tuple(row) == most_common_row,
axis=1).index.tolist()

    // Calculate the ratio of max count to total sequences
    max_count_ratio <- max_count / total_sequences

    events_idx[event] <- mc_indices // Store the event indices
    pattern_lengths.append(len(event)) // Record the length of the pattern
    frequencies.append(max_count_ratio) // Store frequency of most common pattern
  EndFor

  Return events_idx, pattern_lengths, frequencies
EndProcedure

```

Algorithm 1. A step-by-step representation of the proposed sequence mining workflow.

Runs Prediction

The proposed model relies on a convolutional neural network architecture. The input layer is of shape 11×10 , followed by two convolutional blocks, each including a max pooling layer. The first block applies 50 filters with a kernel size of 5×5 , and stride of 1. The second block applies 25 filters with a kernel size of 3×3 , and stride of 1. Following the convolutional operations a dropout layer precipitates a flattening layer, and three fully connected layers of sizes 128, 64, and 32 neurons respectively. A final dropout layer is placed before the prediction layer, where a softmax function provides probabilities for each of the two classes. The model's loss is calculated as binary cross-entropy. The entire architecture can be seen in Figure 3.

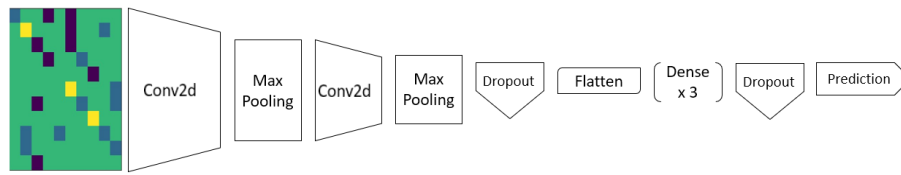


Fig 3. Model Architecture

In order to benchmark the model's performance, several other classifiers were trained, so as to provide comparisons. Namely, the data was used to train a Support Vector Classifier, a Decision Tree Classifier, a Random Forests algorithm, and a k-Nearest Neighbors algorithm. The classifiers were evaluated using a 10-fold cross-validation approach. Although the original play-by-play data are inherently sequential, our modeling framework operates on transformed event matrices that represent independent segments of play rather than continuous time series. Each input instance corresponds to an 11×10 matrix encoding ten consecutive events and their associated features preceding (or not preceding) a scoring run. These matrices are flattened and sampled independently, ensuring that no temporal overlap occurs between training and testing samples.

Because the model does not rely on temporal continuity across sequences, but instead learns from self-contained event windows, standard 10-fold cross-validation is an appropriate validation strategy. This approach enables robust estimation of generalization performance without introducing data leakage or bias from temporally dependent samples. The setup aligns with best practices for independent, event-based prediction tasks, rather than time-series forecasting frameworks that require ordered or blocked validation schemes.

Sequence Generation

For this task, we will be using our trained RunNet model and approaching it as an optimization problem. The goal is that given partial sequences, our model will be generating the rest of the input so that the entire sequence can be classified as a run. We are defining a loss function that attempts to maximize the output probability of 'run':

$$L = -\log(P('run' \vee x_{full}) + \epsilon)$$

Where:

- $P('run' \vee x_{full})$ is the predicted probability for the 'run' class given the full input x_{full} .
- $x_{full} = [x_{partial}, x_{missing}]$ is the complete input, consisting of both the partial known input $x_{partial}$ and the generated missing input $x_{missing}$.
- $\epsilon = 1 \times 10^{-8}$ to prevent $\log(0)$.

Using the above loss function, we attempt to estimate $x_{missing}$ so as to minimize L .

We are also interested in measuring the confidence of the optimization algorithm for the missing part of the input. Essentially, we are measuring the reliability of existing evidence. This is expected to naturally increase as more data becomes available. Therefore, at the same time we define:

$$\text{confidence} = \frac{1}{1 + \alpha \cdot \text{loss_value}} \times \left(\frac{i}{\text{max_len}} \right)$$

Where:

- *confidence*: The computed confidence score.
- *loss_{value}*: The loss value for the current prediction.
- *i*: The length of the partial input sequence being evaluated.
- *max_{len}*: The maximum possible length of the partial sequence, up to 9.
- α : A hyperparameter controlling the influence of the loss value on the confidence score. A lower value results in the calculation being more reliant on the length of the partial input, while a higher value puts more weight into the loss.

Results

Focused Sequence Mining

The results of the focused sequence mining process are summarized in Table 4 and Table 5. Each table lists the most frequent event sequences identified in the play-by-play data that precede the initiation of a scoring run. In the tables, ‘S’ indicates that the action was performed by the same team that initiated the run, while ‘O’ denotes an action taken by the opponent.

For each sequence of length L , we report the absolute frequency (Freq), the second-highest frequency (2nd Freq) for comparison, and the Ratio, which represents the proportion of occurrences of that sequence among all pre-run sequences of the same length. These metrics provide a clear view of how often specific combinations of in-game events occur immediately before a run, allowing us to identify patterns that may signal the onset of team momentum.

The sequences were generated using our focused sequence mining algorithm, which extracts frequent event combinations directly from the play-by-play dataset. The results reveal that certain event patterns—such as consecutive successful possessions or defensive stops—consistently precede scoring runs.

Table 4. Home Rules: The table demonstrates the frequency of the most common sequences mined per length. Sequence length indicates how many events are contained in the sequence. 'Freq' and '2nd Freq' indicate the number of occurrences of the most common and second most common sequence per length. Similarly, 'Ratio' and '2nd Ratio' present their relative frequency. Finally, 'Rules' demonstrate the specific events that the most common sequence is comprised of.

Sequence Length	Freq	2nd Freq	Ratio	2nd Ratio	Rules
Last	4737	2577	42.1%	23.1%	DefFoul(S)→run
Last 2	566	438	10.6%	8.2%	CloseShot(O)→DefFoul(S)→run
Last 3	142	104	5.0%	3.7%	DefReb(O)→3ptShot(O)→ DefReb(S)→run
Last 4	61	42	3.3%	2.3%	3ptShot(S)→DefReb(O)→ 3ptShot(O)→DefFoul(S)→run
Last 5	19	16	1.3%	1.1%	3ptShot(S)→DefReb(S)→ 3ptShot(O)→DefReb(S) →OffFoul(S)→run

Table 5. Away Rules: The table demonstrates the frequency of the most common sequences mined per length. Sequence length indicates how many events are contained in the sequence. 'Freq' and '2nd Freq' indicate the number of occurrences of the most common and second most common sequence per length. Similarly, 'Ratio' and '2nd Ratio' present their relative frequency. Finally, 'Rules' demonstrate the specific events that the most common sequence is comprised of.

Events	Freq	2nd Freq	Ratio	2nd Ratio	Rules
Last	4135	2304	41%	22.7%	DefFoul(S)→run
Last 2	107	88	2.2%	1.9%	3ptShot(O)→DefReb(S)→run
Last 3	84	65	3.1%	2.4%	DefFoul(O)→FTShot(S)→ OffReb(S)→run
Last 4	3	2	0.15%	0.1%	3ptShot(S)→DefReb(O)→ MidShot(O)→DefFoul(S)→run
Last 5	2	2	0.17%	0.17%	3ptShot(S)→DefReb(S)→ CloseShot(S)→OffReb(S) →DefFoul(O)→run

Runs Prediction

The model was set to be trained for 20 epochs, using a root mean square propagation algorithm for optimization. An early stopping callback was set up to monitor validation accuracy, which halted the training process after 15 epochs. In Figure 4 we demonstrate the training process performance, while in Table 6 we present metrics on its predictive performance and compare the proposed model to traditional classifiers trained on the same task. Significantly, the model boasts noticeable ability in identifying sequences that lead to runs, while the overall accuracy is 81%. Importantly, our approach appears to handle this classification task at a more favourable level.

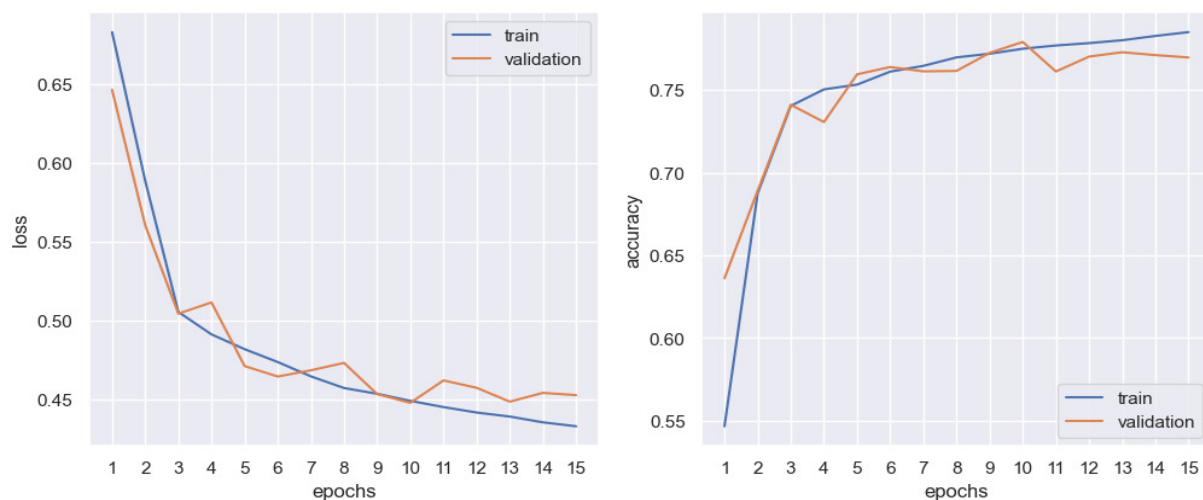


Fig 4. Calculated Loss & Accuracy

Table 6. Traditional Classifiers Performance

Classifier	Accuracy	Precision	Recall	f1-score
RunNet	0.81	0.86	0.81	0.81
Random Forest	0.79	0.78	0.78	0.78
Support Vector	0.76	0.73	0.73	0.73
Decision Tree	0.73	0.73	0.73	0.73
k-Nearest Neighbors	0.59	0.59	0.59	0.59

Features Importance

We aim to further explore the two dimensions of our patterns, both temporal and feature-wise. To that end we investigate how our proposed model learns from the matrices constructed. We compute the gradient of the run class score with respect to the input image, where each pixel's intensity reflects how sensitive the output is to changes in that pixel. The magnitude of the gradient indicates the degree of influence a pixel or region has on the network's decision-making process. Thus, we create a saliency map that helps us understand the decision-making process of our model, by providing insight into the spatial patterns and features that drove the network's predictions. In Figure 5 we can see the aggregated saliency map. We demonstrate two points of interest. First, that the model tends to learn more from more recent events. Secondly, the 'Shooter' feature appears to be more prominent throughout the temporal dimension. Finally, the 'Fouler' feature in the last event is shown to be the most important for the initiation of a scoring run. This specific feature receives values for both offensive and defensive fouls. This, in turn, implies that fouls committed close to the initiation of a scoring run can be vital, regardless of perpetrator, as the model tends to assign particular weight to the feature in either case.

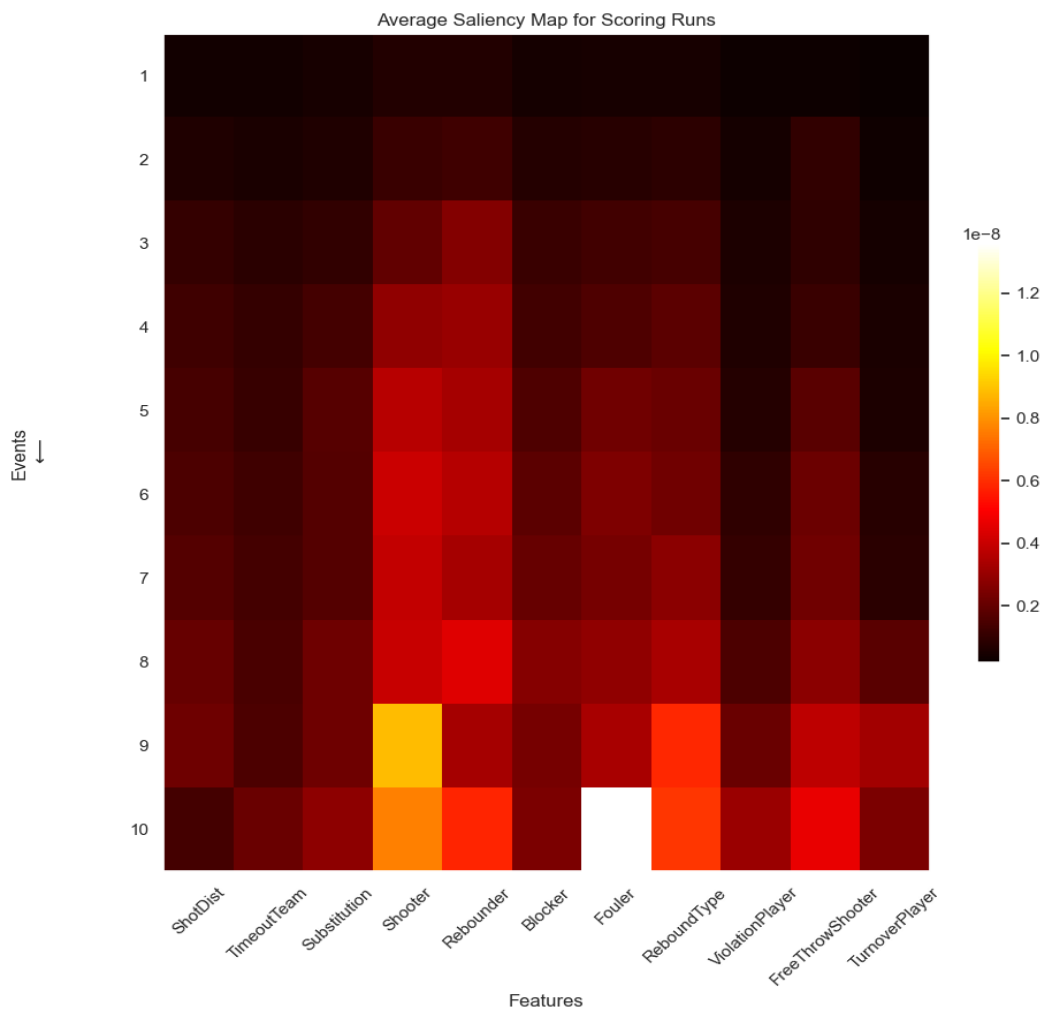


Fig 5. Saliency Map of RunNet's focus on input features. Event 1 indicates the event at the start of the pre-run sequence, while Event 10 indicates the event right before the run.

Sequence Generation

After running the optimization algorithm on 500 samples, we obtain predictions for various

partial lengths in the range [1,9]. In Table 7, we present the mean accuracy score of the predictions of our model compared to ground truths of only 'run' labels, as well as the confidence value at the $\alpha = 1$ level.

Table 7. Optimization Results for 'run' class

Partial Length	Accuracy	Confidence
1	0.94	0.10
2	0.92	0.20
3	0.86	0.30
4	0.88	0.40
5	0.75	0.50
6	0.72	0.60
7	0.69	0.70
8	0.49	0.80
9	0.65	0.85

Discussion

The current study provides significant insights into the mechanics of scoring runs in basketball, a critical factor often cited as a key determinant in game outcomes. By utilizing convolutional neural networks (CNNs) to recognize patterns in play-by-play data, this research marks a notable advancement in sports analytics, particularly in basketball. The results obtained underscore the effectiveness of pattern recognition models for this task. These findings contribute to a growing body of literature aimed at understanding real-time game dynamics and offer substantial implications for both coaching strategies and in-game decision-making processes.

One of the most striking outcomes of this research is the ability of the CNN model to predict the onset of scoring runs by identifying recurring sequences in play-by-play data. Previous studies, as highlighted in the literature review, have predominantly focused on longer-term game or season outcomes, with a few exceptions addressing short-term game dynamics using different methodologies, such as deep reinforcement learning models. However, none of these studies have directly addressed scoring runs with such precision. The ability to predict scoring runs in real-time can provide a competitive advantage to coaching staff, allowing for timely interventions—such as strategic timeouts or lineup changes—that could potentially disrupt an opponent's momentum.

Focused Sequence Mining

In the sequence mining process, the study successfully identified the most common event sequences that precede scoring runs for both home and away teams. The method relied on analyzing sequences of ten events leading up to the start of a run, which were extracted from the play-by-play data across five NBA seasons. The distinction between home and away sequences provided additional insight into how different game dynamics might unfold based on the location of the game.

The most frequent sequence for home runs involved missed 3pt shots and defensive rebounds by the team starting the run. This suggests that when the opponent is struggling offensively, it creates opportunities for scoring, particularly through successful rebounds and shots from a distance. Away runs seemed to indicate similar patterns.

Interestingly, the study also found that there does not seem to be a significant difference in sequences frequency between home and away teams, further supporting the notion that scoring runs are irrelevant to the venue. As such, the influence of external factors, such as the psychological pressure on home teams or the momentum-building potential of away teams under certain game conditions appear to be balanced in that regard. Additionally, it was demonstrated that there appears to be a positive correlation between a team's final standing and the number of times the most common sequence was repeated. This finding is consistent over the seasons investigated. The identification of these sequences offers a new layer of understanding in how specific in-game events can signal the onset of a scoring run and suggests that game location and external pressures may play a role in shaping these dynamics.

Runs Prediction

The CNN model trained on the play-by-play data demonstrated impressive accuracy in predicting scoring runs, with an overall accuracy of 81%. The model's architecture, which relied on extracting patterns from sequences of game events, was well-suited to the task of recognizing the intricate and often subtle relationships between player actions, shot types, and game context that led to scoring runs. The performance of the CNN model was benchmarked against several traditional classifiers, including random forests, decision trees, and support vector machines. The results showed that RunNet outperformed these alternative models, particularly in terms of precision.

This finding is significant because it underscores the effectiveness of deep learning approaches, particularly CNNs, in handling sequential and temporal data in the context of sports analytics. The success of the model in predicting runs suggests its potential for real-time applications, where identifying and responding to the initiation of a run could have direct implications for coaching strategies during a game.

Features Importance

The analysis of feature importance provided valuable insights into which game elements most strongly influenced the model's predictions. By constructing a saliency map, the study revealed that the model placed greater emphasis on the more recent events leading up to a run, with features related to the shooter, fouls, and shot type being particularly prominent.

The "shooter" feature consistently emerged as the most influential, especially when considering the final few events before a run. This finding is consistent with basketball strategy, where key shooters often play a pivotal role in initiating or maintaining momentum for their team. The saliency map further showed that defensive actions, such as fouls committed by the opponent, also played a critical role in predicting runs. This aligns with the notion that defensive breakdowns or mistakes often provide opportunities for the opposing team to capitalize and build momentum.

Moreover, the temporal aspect of feature importance highlights that the most recent actions carry more weight in determining the likelihood of a run. This suggests that the momentum shift leading to a scoring run is not necessarily built over a long series of events, but rather, can be triggered by a few decisive plays within a short span. Understanding these feature dynamics could provide coaching staff with actionable insights, particularly in how they respond to critical moments in the game.

Sequence Generation

The sequence generation task aimed to complete partial event sequences to predict whether the full sequence would lead to a run. The optimization algorithm developed for this purpose

successfully generated missing parts of sequences, with varying levels of accuracy depending on the length of the partial input. Interestingly, the model demonstrated its highest accuracy (94%) when generating sequences from only one partial event. However, as the length of the partial input increased, accuracy gradually declined, reaching 65% when nine events were observed.

The observed divergence between the declining predictive accuracy and the increasing confidence score is a direct consequence of the solution space constraints and the specific mathematical definition of our confidence metric.

The sequence generation is formulated as an optimization task. When the partial input is short, the model possesses high generative autonomy, allowing it to synthesize an idealized trajectory that aligns perfectly with the 'run' class, resulting in high accuracy (94%). However, as the observed sequence lengthens to nine events, the solution space becomes constricted. The optimization algorithm is forced to align with a rigid, pre-existing historical trajectory, which may naturally lean towards a non-run, making it significantly harder to steer the sequence toward the target class. This constraint leads to the observed drop in accuracy to 65%.

Conversely, the rise in confidence is intrinsic to our metric design. The metric includes a multiplicative term for normalized length to penalize predictions based on scant evidence. Consequently, at short lengths, confidence is capped at 0.1 regardless of optimization success. As the sequence lengthens, the weight of the evidence term increases (0.1→0.9), overpowering the increase in optimization loss. The metric is designed to report low confidence when information is scarce, and higher confidence when the input is information-rich, even if the optimization task itself becomes more difficult.

This capability highlights the model's potential utility in scenarios where incomplete information is available. For example, coaching staff could input partial game sequences during a timeout or between quarters to assess the likelihood of a run occurring based on their current strategy. The confidence metric, which grew with the length of the partial sequence, provides a valuable measure of the reliability of the model's predictions, especially in contexts where rapid decision-making is essential.

The implications of sequence generation for real-time decision-making in sports are particularly promising. Teams could potentially optimize their lineups or play strategies by simulating different outcomes based on incomplete sequences, maximizing their chances of initiating a run or, conversely, making defensive adjustments to prevent an opponent from gaining momentum. Additionally, the model's ability to handle complex scenarios opens opportunities for post-game analysis, where coaching staff can evaluate which sequences likely contributed to key momentum shifts and refine their strategies accordingly.

Overall, this study offers a novel approach to understanding and predicting scoring runs in basketball, with promising applications for both real-time game management and broader sports analytics. The integration of CNNs for pattern recognition in sequential game data has proven to be an effective method, with significant potential for future development. Expanding the feature set, improving the model's generalizability, and applying these techniques to other sports or contexts will further enhance our understanding of momentum and scoring dynamics. Finally, the prospect of real-time applications in coaching and strategy decision-making opens exciting new avenues for data-driven approaches in sports management.

API

During this work, we developed an API that allows a user to leverage the algorithms and models explored. A user can upload their own datasets and extract rules for runs initiation

through focused sequence mining. Additionally, they can train their own models on that dataset for runs predictions. Finally, users can generate full sequences out of partial inputs. The API can be found as part of the repository linked in the 'Code Availability' section.

Limitations

Despite the promising results, there are limitations to this study that warrant consideration. First, the reliance on play-by-play data alone may not capture the full complexity of basketball games. While the use of 11 features provided a robust framework for analysis, other contextual factors—such as player fatigue, psychological states, or crowd influence—were not included. Future research could incorporate additional data sources, such as player tracking systems or biometric data, to further enhance the accuracy and applicability of the models. Furthermore, the current model is trained on historical data from NBA games between 2015 and 2020, which, while extensive, might not fully generalize to other basketball leagues or eras. Expanding the dataset to include games from different leagues, such as EuroLeague or collegiate basketball, could provide a more comprehensive understanding of the phenomenon.

Another potential limitation lies in the definition of a scoring run. The study confines a scoring run to a sequence of four consecutive baskets made by one team without the opponent scoring, which may not capture all relevant instances of momentum shifts. In some cases, shorter or longer runs might also be significant, particularly when considering the emotional and psychological aspects of the game. Moreover, external factors like TV timeouts or fouls, which have been shown to influence game flow, could also affect the onset or interruption of runs. A more dynamic and nuanced approach to defining runs could be explored in future studies.

The performance of alternative classifiers, such as random forests or support vector classifiers, as discussed in the results section, also highlights the complexity of predicting scoring runs. Although CNNs outperformed traditional classifiers in this context, further experimentation with other machine learning models, such as attention-based networks or ensemble methods, may yield even better results. Additionally, while the current model achieved a high level of accuracy, it is important to consider the practical implications of false predictions in real-time settings. A false prediction could lead to misguided strategic decisions, such as premature timeouts or unnecessary substitutions, potentially disrupting a team's rhythm rather than aiding it.

Finally, we address the sensitivity of our 'run' definition. While our current approach effectively captures unanswered scoring sequences, allowing for offensive misses provided the opponent is held to zero points, it adheres to a strict 'shutout' constraint. This definition provides a high-signal ground truth for the optimization task but may exclude organic, high-momentum periods where the opponent manages a single isolated score (e.g., a '10-2 run' rather than a '10-0 run').

Future iterations of this work could explore relaxing this constraint in favor of a 'Net Scoring Advantage' definition. By allowing for minor defensive lapses within a broader dominant stretch, the model could be extended to detect a wider variety of momentum shifts.

Conclusions & Future Work

In this study, we developed a novel approach for identifying and predicting scoring runs in basketball games using pattern recognition techniques, particularly convolutional neural networks. Our model demonstrated significant accuracy, reaching 81%, in identifying sequences of events that lead to scoring runs. By analyzing play-by-play data from five NBA

seasons (2015–2020), we were able to discern key features and sequences that precede runs, providing valuable insights into how teams gain momentum. Additionally, we developed a generative model that can complete partial sequences and predict the likelihood of a run based on incomplete data, offering potential real-time applications for coaching and strategy adjustments.

This study marks a substantial contribution to the field of sports analytics by shifting the focus from long-term game or season outcomes to more immediate and actionable game dynamics. The pattern recognition methodology we introduced, combined with deep learning models, offers a fresh perspective on understanding how specific actions and game states influence momentum shifts during a basketball game. Our results underscore the potential of machine learning tools in sports analytics, particularly in aiding coaching decisions through real-time data analysis and predictive modeling.

In the future, this research can be expanded by incorporating additional features such as player fatigue, crowd noise, and detailed player tracking data to enhance prediction accuracy and gain a more comprehensive understanding of the factors influencing scoring runs. Additionally, optimizing the model for real-time implementation is crucial for practical use during live games, reducing computational costs and latency. Exploring alternative architectures like autoencoders or attention-based models may further improve the model's handling of sequential and temporal data. Moreover, applying the current methodology to other sports where momentum plays a critical role, such as soccer or hockey, could open new areas of research in sports analytics. Lastly, future studies could investigate predictive interventions, using models to suggest strategic decisions like timeouts or substitutions that could disrupt or initiate scoring runs, providing real-time data-driven support for coaching staff.

Code Availability

The entire code developed for this work is available [here](#). Additionally, we provide an API allowing for uploading datasets, sequence mining, training a model and using it for predictions, and, finally, for sequence optimization.

References

- Agrawal, R., & Srikant, R. (1995). Mining sequential patterns. *Proceedings of the Eleventh International Conference on Data Engineering*, 3–14. <https://doi.org/10.1109/ICDE.1995.380415>
- Bunker, R., & Susnjak, T. (2022). The application of machine learning techniques for predicting match results in team sport: A review. *Journal of Artificial Intelligence Research*, 73, 1285–1322. <https://doi.org/10.1613/jair.1.13509>
- Bunker, R. P., & Thabtah, F. (2019). A machine learning framework for sport result prediction. *Applied Computing and Informatics*, 15(1), 27–33. <https://doi.org/10.1016/j.aci.2017.09.005>
- Cheng, H., Yan, X., Han, J., & Yu, P. S. (2008). Direct discriminative pattern mining for effective classification. *2008 IEEE 24th International Conference on Data Engineering*, 169–178. <https://doi.org/10.1109/ICDE.2008.4497425>
- Claudino, J. G., Capanema, D. D. O., de Souza, T. V., Serrão, J. C., Machado Pereira, A. C., & Nassis, G. P. (2019). Current approaches to the use of artificial intelligence for injury risk assessment and performance prediction in team sports: a systematic review. *Sports medicine-open*, 5(1), 28. <https://doi.org/10.1186/s40798-019-0202-3>

- Decroos, T., Bransen, L., Van Haaren, J., & Davis, J. (2019). Actions speak louder than goals: Valuing player actions in soccer. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1851–1861. <https://doi.org/10.1145/3292500.3330758>
- Fernández, J., Bornn, L., & Cervone, D. (2021). A framework for the fine-grained evaluation of the instantaneous expected value of soccer possessions. *Machine Learning*, 110(6), 1389–1427. <https://doi.org/10.1007/s10994-021-05989-6>
- Han, J., Pei, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., & Hsu, M. (2001). Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. *Proceedings of the 17th International Conference on Data Engineering*, 215–224. <https://doi.org/10.1109/ICDE.2001.914830>
- Horvat, T., & Job, J. (2020). The use of machine learning in sport outcome prediction: A review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(5), e1380. <https://doi.org/10.1002/widm.1380>
- Iso-Ahola, S. E., & Dotson, C. O. (2014). Psychological momentum: Why success breeds success. *Review of General Psychology*, 18(1), 19–33. <https://doi.org/10.1037/a0036406>
- Ji, Y., Ye, G., & Cheng, H. (2014). Interactive body part contrast mining for human interaction recognition. *2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 1–6. <https://doi.org/10.1109/icmew.2014.6890714>
- Karthikeyan, V., & Priyadharsini, S. S. (2024). A stacked convolutional neural network framework with multi-scale attention mechanism for text-independent voiceprint recognition. *Pattern Anal. Appl.*, 27(2), 48. <https://doi.org/10.1007/S10044-024-01278-9>
- Klaassen, F., & Magnus, J. R. (2014). *Analyzing Wimbledon: The power of statistics*. Oxford University Press. <https://doi.org/10.1093/acprof:oso/9780199355952.001.0001>
- Li, C., Zhang, H., Zhang, Y., Shen, J., & An, R. (2025). The application of artificial intelligence techniques in predicting game outcomes of professional basketball league: A systematic review. *Plos one*, 20(6), e0326326. <https://doi.org/10.1371/journal.pone.0326326>
- Livieris, I. E., Pintelas, E., & Pintelas, P. (2020). A CNN–LSTM model for gold price time-series forecasting. *Neural Computing and Applications*, 32, 17351–17360. <https://doi.org/10.1007/s00521-020-04867-x>
- Mannila, H., Toivonen, H., & Inkeri Verkamo, A. (1997). Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1(3), 259–289. <https://doi.org/10.1023/A:1009748302351>
- Maymin, P. Z. (2017). The automated general manager: Can an algorithmic system for drafts, trades, and free agency outperform human front offices? *Journal of Global Sport Management*, 2(4), 234–249. <https://doi.org/10.1080/24704067.2017.1389248>
- Meng, M., Zhang, Y., Ma, Y., Gao, Y., & Kong, W. (2023). EEG-based emotion recognition with cascaded convolutional recurrent neural networks. *Pattern Anal. Appl.*, 26(2), 783–795. <https://doi.org/10.1007/S10044-023-01136-0>
- Morgulev, E., Voslinsky, A., Azar, O. H., & Bar-Eli, M. (2020). Biased perceptions about momentum: Do comeback teams have higher chances to win in basketball overtimes? *Judgment and Decision Making*, 15(4), 545–560. <https://doi.org/10.1017/S1930297500007488>
- Ötting, M., Langrock, R., & Maruotti, A. (2023). A copula-based multivariate hidden markov model for modelling momentum in football. *ASTA Advances in Statistical Analysis*, 107(1), 9–27. <https://doi.org/10.1007/s10182-021-00395-8>

- Perin, C., Vuillemot, R., Stolper, C. D., Stasko, J. T., Wood, J., & Carpendale, S. (2018). State of the art of sports data visualization. *Computer Graphics Forum*, 37, 663–686. <https://doi.org/10.1111/cgf.13447>
- Rico-González, M., Pino-Ortega, J., Méndez, A., Clemente, F., & Baca, A. (2023). Machine learning application in soccer: a systematic review. *Biology of sport*, 40(1), 249–263. <https://doi.org/10.5114/biolSport.2023.112970>
- Ryall, E., & Edgar, A. (2022). Watching sport during COVID-19. In *Philosophy, sport and the pandemic* (pp. 139–151). Routledge. <https://doi.org/10.4324/9781003214243-12>
- Severini, T. A. (2020). *Analytic methods in sports: Using mathematics and statistics to understand data from baseball, football, basketball, and other sports*. Crc Press. <https://doi.org/10.1201/9780367252090>
- Stival, L., Pinto, A., Andrade, F. D. S. P. D., Santiago, P. R. P., Biermann, H., Torres, R. D. S., & Dias, U. (2023). Using machine learning pipeline to predict entry into the attack zone in football. *PloS one*, 18(1), e0265372. <https://doi.org/10.1371/journal.pone.0265372>
- Sun, C., Wu, G., Pan, G., Zhang, T., Li, J., Jiao, S., Liu, Y.-C., Chen, K., Liu, K., Xin, D., & Gao, G. (2024). Convolutional neural network-based pattern recognition of partial discharge in high-speed electric-multiple-unit cable termination. *Sensors*, 24(8), 2660. <https://doi.org/10.3390/S24082660>
- Tang, Q., Wei, X., & Tan, B. (2026). The Role of Machine Learning in Talent Identification for Team Sports: A Systematic Review. *Journal of Sports Science and Medicine*, 25(1), 58–83. <https://doi.org/10.52082/jssm.2026.58>
- Xarles, A., Escalera, S., Moeslund, T. B., & Clapés, A. (2025). Action Valuation in Sports: A Survey. In *Proceedings of the Computer Vision and Pattern Recognition Conference* (pp. 6132–6142). <https://doi.org/10.48550/arXiv.2504.06163>
- Xu, X., Liu, P., & Guo, M. (2023). Drainage pattern recognition of river network based on graph convolutional neural network. *ISPRS Int. J. Geo Inf.*, 12(7), 253. <https://doi.org/10.3390/IJGI12070253>
- Yin, W., Kann, K., Yu, M., & Schütze, H. (2017). Comparative study of CNN and RNN for natural language processing. *arXiv Preprint arXiv:1702.01923*. <https://doi.org/10.48550/arXiv.1702.01923>
- Zaki, M. J. (2001). SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1), 31–60. <https://doi.org/10.1023/A:1007652502315>