An Explainability Focused Computational Pipeline for Anti-Cheat and Churn Monitoring in Online Gaming Platforms

Dr B Vani[1], Bharath R[2], Meghana P[3], Sake Ajay[4], Srinith N[5]

[1]*H.O.D, Sambhram Institute of Technology, Bengaluru*

[2,3,4,5]*Student, Sambhram Institute of Technology, Bengaluru*

Abstract—**Online Multiplayer games face regular difficulty related to cheating and player churn, both of which negatively impact Equality, engagement and platform sustainability. Existing detection mechanisms often rely on single-modal analysis or operate as black-box systems, limiting adaptability and trust. This paper presents an explainable multi-view computational framework that jointly addresses cheating detection and churn prediction by integrating player telemetry features, behavioral patterns, social interaction structures, and gameplay images. Four complementary views are included in the proposed system: (i) an image-based view that uses convolutional neural networks to detect visual cheating artefacts; (ii) a behavioural view for churn prediction based on temporal engagement indicators; (iii) a social view that models player interactions as networks to identify influence-driven risk; and (iv) a portrait view that uses supervised machine learning on aggregated player features. Using SHAP for tabular models, network-based visual reasoning for social analysis, and Grad-CAM for CNN-based image predictions, explainability is integrated throughout the pipeline. The framework is implemented as a Flask-based web application with interactive dashboards, persistent logging, and real-time inference. Strong performance is demonstrated by experimental evaluation across all modules, with high detection accuracy, low false-positive rates, and understandable explanations for humans. The findings show that explainable AI in conjunction with multi-view learning offers a reliable and practical method for contemporary online game analytics.**

Index Terms— **SHAP, Grad-CAM, CNN, explainable AI, multi-view learning, player churn prediction, online game cheating detection, and machine learning.**

I. INTRODUCTION

Online multiplayer games have evolved into composite digital systems that produce large number of behavioral, interactional, and visual data through continuous activity [1].

Due to the availability of such data, game analytics has emerged as a distinct field of study with the goals of comprehending player behaviour, enhancing engagement, and preserving fair gaming environments [1]. However, the growing popularity and scope of online games have created serious security issues, especially when it comes to stopping unfair practices that compromise game integrity [2]. Cheating remains a persistent and evolving problem in online games, surrounding a wide range of behaviors such as automation, information contact, and rule manipulation [3]. These activities not only disrupt competitive balance but also degrade player trust and overall game experience. Behavioral analysis has been widely explored as an effective approach for detecting cheating players by identifying abnormal gameplay patterns [4]. In particular, anomaly detection techniques have shown promise in identifying previously unseen cheating behaviors without relying on predefined rules [5]. The thing about machine learning is that it is getting better and better. Because of this people are using deep learning models more and more to catch cheaters in online games. Some studies have already shown that machine learning is really good at finding cheaters by looking at what players do. Machine learning classifiers can use things like player features to identify cheaters. Recently people have started using deep learning and special models to look at what players do over time. This helps catch cheating patterns in the logs of player activity. These methods are really good, at catching cheaters. They are not very transparent. This means that people do not always trust machine learning models to catch cheaters in games because they do

not know exactly how they work. Machine learning is used to detect cheaters in games, and it is continually improving at this task. In addition to cheating detection, player churn has become a significant concern for online game developers, particularly in free-to-play environments where long-term engagement has a direct impact on the profits [12]. Churn prediction models usually look at how people behave, like how they play, how long they play for and when they stop playing to figure out if a player is going to stop playing. If we can find out who is likely to stop playing on we can try to keep them playing with special strategies. The problem is that the old models that try to predict when people will stop playing do not tell us why people are stopping, they just tell us that they will stop. Some new research says that we need to make intelligence that can explain itself especially when it comes to game analytics so we can understand what is going on with the complex machine learning models that are used to predict player churn. Player churn is a deal and we need to understand it better. Explainability techniques such as SHAP enable feature-level understanding of tabular models, while visual explanation methods like Grad-CAM provide insight into convolutional neural network decisions for image-based tasks [19], [21]. These techniques are essential for supporting ethical decision-making, reducing false accusations, and enabling human-in-the-loop moderation systems [11]. In this paper, an explainable multi-view framework for detecting cheating and predicting player churn in online games is presented.
The new system combines four different, but equally important, sets of data analyses:(i) a portrait view that represents the features of players aggregated, (ii) a behavioral view for the prediction of churn by using patterns of engagement, (iii) a social view that represents the interactions of players as networks, and (iv) an image, based view that uses convolutional neural networks to recognize the visual cheating signals. Moreover, the framework employs SHAP, social network visualisation, and Grad, CAM for explanation purposes, thus, maintaining openness and trust. The framework is a real, end, to, end, web, based system, which has been put into operation and tested with different datasets, thereby proving its performance and the ability to be understood in the gaming world.

Background and Motivation

Game analytics has turned into a major subject of investigation that helps to understand player behavior, enhance game design, and secure large, scale online multiplayer environments [1]. Contemporary games produce a variety of data that comprises gameplay logs, interaction networks, and visual content and can be used to spot the occurrence of the malicious activities and to ensure fair play. Online game cheating has been divided in a comprehensive manner depending on the purpose and the level of the performer; thus the classification embraces the behaviors automation, information exposure, and rule manipulation [3].

Conventional rule, based and signature, based anti, cheat methods are leaking ways of tricking and require by hands frequent updates so they cannot be the powerful weapons against the already evolved cheating strategies [2]. Most of the currently existing methods for detecting cheating and for predicting churn are single, view data sources such as gameplay statistics or behavioral logs [4]. These models, which are effective in controlled environments, are usually incapable of detecting coordinated or socially driven behaviors and may even overlook peer influence on player disengagement [13]. The study has indicated that analysis of the social network might uncover the cheating rings and the coordinated malicious activities through the player interaction graphs [10]. Along with that, the deep learning-based image processing has been specified as a potent method in detecting the visual cheating cues that are invisible to the behavioral data [7].

II. LITERATURE REVIEW

Research on cheating detection, player churn prediction, and explainable artificial intelligence (XAI) in online games has advanced significantly over the past decade, evolving from heuristic-based methods to data-driven, multi-view, and interpretable machine learning frameworks.

A. Cheating Detection Methods

Early research on cheating detection primarily focused on behavioral anomaly detection. Nguyen et al. Proposed unsupervised techniques used clustering and statistical deviation analysis to identify unusual gameplay patterns without labeled data [5]. Subsequent studies compared distance-based, density-based, and isolation-based anomaly detectors. These studies highlighted their sensitivity to the characteristics of datasets and the high false-positive rates in changing gaming environments [4]. These limitations led to a shift toward supervised learning methods. Supervised machine learning methods showed better performance by using engineered player telemetry features. Willman demonstrated that classical classifiers like Random Forests and Support Vector Machines could effectively identify cheating players when trained on labeled behavioral data [6]. However, these models depend heavily on annotated datasets and have difficulties generalizing to new cheat strategies. Deep learning techniques further improved this field by modeling the timing and sequence of player behavior. Pinto et al. applied deep learning to multivariate time-series representations of gameplay interactions, enabling detection of complex cheating behaviors such as aimbots and trigger bots [7]. More recent work explored automated cheating detection using machine learning pipelines, confirming their effectiveness across diverse game genres [8].

Foundational contributions in this domain include the systematic taxonomy of cheating behaviors introduced by Yan and Randell, which continues to guide modern cheat classification [3], as well as behavioral bot detection using fine-grained player action analysis [9]. Socially coordinated cheating has also gained attention, with recent human-in-the-loop frameworks demonstrating improved detection of organised cheating groups through expert-guided model refinement [11].

B. Explainable AI for Cheating Detection

As machine learning models became increasingly complex, explainability emerged as a critical requirement for trustworthy cheating detection. Tao et al. They introduced a clear and understandable framework for cheating detection that combines behavioral, social, and visual aspects using SHAP, Grad-CAM, and graph-based explanations [17]. Their earlier work set the stage for multi-view cheating detection by showing how to incorporate explainability across different types of data [18]. Key explainability techniques that support this area of research include SHAP for feature-level attribution in tabular models [19], consistent feature attribution for tree-based ensembles [20], and Grad-CAM for visual explanations of convolutional neural network predictions [21]. These techniques improve transparency, support auditing, and allow for human oversight by explaining why a player was flagged, rather than just providing a yes or no decision.

C. Player Churn Prediction and Explainability

Player churn prediction has been studied a lot because it is important for keeping players engaged and generating revenue in online games. Early studies used supervised learning methods on behavioral data to predict when players might disengage, especially high-value players [14]. Later, deep learning methods were introduced to better understand non-linear and time-based engagement patterns in churn prediction [15]. Recent research has focused on explainability and the impact of social influence in churn modeling. Mustač et al. examined churn prediction in free-to-play games using behavioral indicators and labelling strategies [12]. Loria et al. demonstrated that meaningful churn prediction can be achieved even with limited behavioral summaries [13]. More recent studies explored clear churn modeling using fuzzy logic and explainable machine learning to uncover transparent churn patterns [22]. Others used explicit and implicit behavioral features to improve interpretability [24]. Social interaction-aware models that use graph neural networks have further improved churn prediction accuracy by capturing peer influence effects [16].

D. Multi-View and Integrated Approaches

Despite significant progress, most previous work treats cheating detection and churn prediction as separate issues. They often rely on just one type of data, like behavioral logs or visual evidence. Only a few studies have looked at unified, multi-view frameworks that combine behavioral, social, and visual signals with clear explanations. Recent research shows the promise of multi-view explainable systems for thorough player analysis. However, many methods still require heavy computation or are more theoretical. The system proposed in this paper overcomes these challenges by combining portrait-level profiling, behavioral churn prediction, social network analysis, and image-based cheating detection in a single, explainable framework. By using SHAP-based feature attribution, social graph visualization, and Grad-CAM explanations, this approach builds on existing multi-view frameworks while providing a lightweight and practical solution suitable for both real-world and academic gaming settings.

III. PROBLEM DEFINITION

Online gaming platforms rely on automated analytics to detect cheating behavior and predict player churn in order to preserve fair gameplay and sustain engagement. While machine learning-based approaches have improved detection accuracy, current solutions still struggle with issues like robustness, transparency, and usability. Many systems, in particular, do not clearly explain why a player is marked as a cheater or predicted to disengage. This limits trust, the effectiveness of moderation, and ethical deployment.

A. Cheating Detection Challenges

Cheating in online games has different types, such as aimbots, ESP overlays, macro automation, bot, driven gameplay, and exploitation of game rules. Conventional anti, cheat measures like rule, based filters, signature matching, and manual review are finding it hard to keep up with the continuously changing cheating strategies. Several contemporary detection systems depend solely on one analytical perspective, such as behavioral telemetry or visual evidence, which limit their capability in uncovering different cheating patterns. Purely behavioral methods may not see visual cheats, while image, based methods may not be able to detect subtle gameplay anomalies. Besides that, anomaly, based detectors frequently have a problem with false, positive rates, wherein they incorrectly identify legitimate players. Additionally, the majority of machine-learning based cheating detection systems that are used as black boxes also have the problem of not providing explanations for their decisions and thus, lowering the trust in automated flagging.

B. Player Churn Prediction Challenges

Player churn prediction aims at figuring out users who are going to stop using so that it is possible to apply retention strategies in time. Nevertheless, churn prediction challenges arise from the fact that engagement patterns are very dynamic and heavily dependent on gameplay modes, updates, and individual preferences. Churn, related behaviors are usually non, linear and become gradually more complex over time, thus static models have low accuracy. Besides that, different definitions of churn may result in unstable classes and therefore model reliability will be low. There are numerous models for churn prediction that only provide risk scores without giving an explanation of the factors that contributed to this, thus their intervention targeted usefulness is limited.

C. Need for a Unified, Explainable Framework

Most of the time, the existing research has different approaches to detecting cheating and predicting churn, seldomly fusing these two tasks and is often single, modal. However, the reality is that player behavior, social interactions, and visual gameplay seem to be highly interdependent. As a result, there is a demand for a consolidated system that simultaneously examines various perspectives of player activity while also keeping the decision, making process transparent. Such a system should not only be capable of cheating detection and churn prediction but also deliver rationales understandable by humans for every player that is flagged. Hence, the problem that this research work is dealing with is the formulation of an explanatory multi, view framework which combines portrait, level features of the players, behavioral patterns of engagement, analysis of social interaction, and image, based evidence to detect cheating and predict churn in a transparent way that is readily deployable. The aim is to have every automated decision be correct and interpretable in conjunction with ethical moderation, player trust, and informed game management.

IV. PROPOSED SYSTEM

This section presents the proposed explainable multi-view framework for detecting cheating and predicting player churn in online gaming platforms. The architecture is designed to be scalable, modular, and interpretable, enabling reliable deployment in real-world gaming environments. To achieve this, the system clearly separates offline model training from online inference and decision-making, allowing computationally intensive learning processes to be performed independently of real-time gameplay operations. Furthermore, the framework integrates an explainability layer that provides human-understandable justifications for model predictions, enhancing transparency, trust, and ethical accountability. A dedicated user-facing interface is also incorporated to visualise predictions, explanation outputs, and risk indicators, while supporting feedback from moderators and administrators. This design ensures that automated decisions remain auditable, actionable, and aligned with practical moderation workflows.

A. System Overview

The proposed system follows a modular architecture comprising four analytical views: portrait, behavioral, social, and image-based views. Each view processes a different modality of player data and contributes independent predictions and explanations. Model training is performed offline using historical data, while real-time predictions and explanations are generated during online inference using pre-trained models.

Fig. 1 illustrates the offline training architecture, and Fig. 2 presents the online inference and user interface architecture of the proposed system.

B. Offline Model Training and Preparation

The offline training phase is responsible for data preprocessing, feature engineering, and model learning for each analytical view This stage uses labeled datasets obtained from public sources and artificially generated data to ensure robustness and class balance.

Aggregated player features are utilized for the portrait view to train a machine learning classifier for cheating detection. The behavioral view leverages temporal engagement features to train a churn prediction model. The social view builds player interaction graphs and extracts graph, based features for social risk analysis. The image, based view trains a convolutional neural network with labelled gameplay screenshots to detect the presence of visual cheating. Each model that has been trained is saved and deployed for use during the online inference phase.

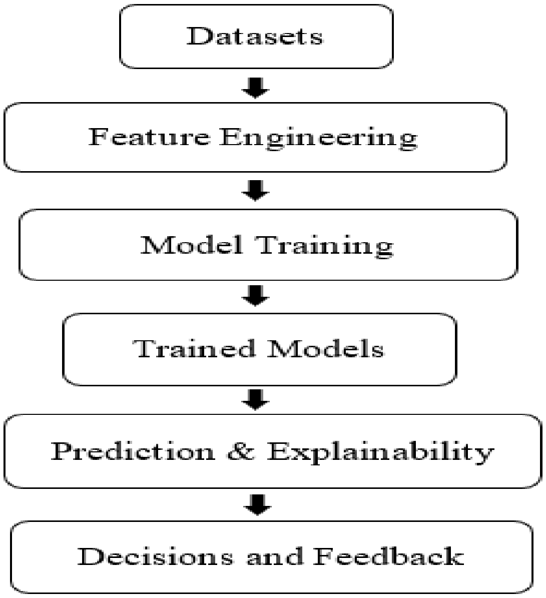

Fig.1. Workflow of the proposed system showing data processing, model training, explainability, and feedback integration.

Algorithm 1: Offline Training Procedure

Input: Portrait dataset Dp, Behavioral dataset Db, Social dataset Ds, Image dataset Di

Output: Trained models Mp, Mb, Ms, Mi

```
1: Preprocess all datasets and handle missing values
2: Perform feature engineering for portrait and behavioral datasets
3: Construct player interaction graph from social dataset
4: Train portrait view classifier $M_p$using $D_p$
5: Train behavioral churn prediction model $M_b$using $D_b$
6: Extract graph-based features and prepare social model $M_s$using $D_s$
7: Train CNN-based image model $M_i$using $D_i$
8: Validate all models on held-out test data
9: Tune hyperparameters using cross-validation
10: Generate explainability artifacts using SHAP and Grad-CAM
11: Evaluate model robustness and consistency across views
12: Store trained models and explanation outlines for deployment
```

C. Online Inference and Decision-Making

During online operation, the system uses the pre-trained models to analyze incoming player data and generate predictions. Each analytical view independently produces a prediction along with an explanation. The portrait and behavioral views use SHAP to identify influential features, the social view visualises network-level risk propagation, and the image-based view applies Grad-CAM to highlight visual regions influencing the prediction. Predictions from each view are presented independently rather than being fused into a single opaque score, allowing human reviewers to evaluate evidence from multiple perspectives.

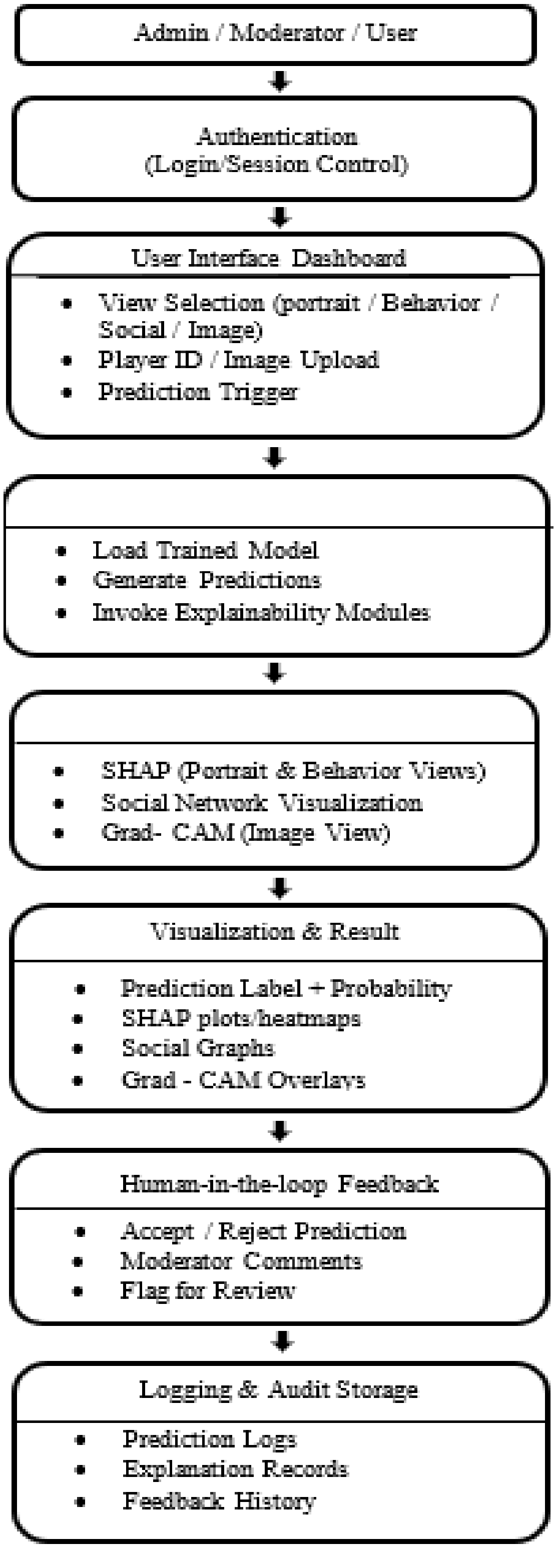

Fig.2. User interface architecture illustrating interaction between moderators, prediction services, explainability modules, and human-in-the-loop feedback in the proposed multi-view framework.

```
Algorithm 2: Online Inference Workflow
Input: Player data xp, xb, xs, xi
Output: Cheat prediction, Churn prediction, View-wise explanations
1: Load pre-trained models Mp, Mb, Ms, Mi
2: Generate portrait view prediction using Mp and xp
3: Generate behavioral view churn prediction using Mb and xb
4: Analyse social risk indicators using Ms and xs
5: Perform image-based cheating detection using Mi and xi
6: Generate SHAP explanations for portrait and behavioral views
7: Generate social graph visualisation for social view
8: Generate Grad-CAM heatmaps for image-based view
9: Display predictions and explanations to the user interface.
```

D. User Interface and Human-in-the-Loop Feedback

The system includes a web-based user interface that presents predictions and explanations from all four views in an interpretable manner. Moderators and analysts can inspect cheating and churn predictions, visualize feature importance and heatmaps, and review social interaction patterns.

A feedback form is provided for each decision, allowing users to validate or contest predictions. This feedback mechanism supports human-in-the-loop review, improves trust in automated decisions, and enables future refinement of the models.

E. Design Rationale

By separating offline training from online inference, the proposed system ensures computational efficiency, scalability, and reproducibility. The modular multi-view design improves robustness against evasion and incomplete data, while integrated explainability and feedback mechanisms ensure that every automated decision is transparent, fair, and justifiable.

V. DATASET DESCRIPTION

The proposed explainable multi-view framework is evaluated using a combination of publicly available datasets and synthetically generated data, covering behavioral, social, and visual perspectives of player activity. This design enables comprehensive analysis while ensuring scalability and reproducibility.

A. Portrait View Dataset

The portrait view dataset represents aggregated player-level characteristics derived from gameplay behavior. This dataset was sourced from publicly available gaming-related datasets hosted on Kaggle and further refined through preprocessing and feature selection. Each record corresponds to an individual player and captures long-term behavioral summaries such as total playtime, win–loss ratio, kill–death ratio, accuracy, average session duration, and activity frequency. To enhance class balance and model robustness, synthetic samples were generated using controlled perturbation and noise injection techniques while preserving statistical consistency with the original data distribution. The portrait view serves as the primary input for global player profiling and baseline cheating risk assessment.

B. Behavioral View Dataset

The behavioral view dataset focuses on temporal engagement patterns for player churn prediction. It consists of time-dependent features such as session frequency, inactivity gaps, login intervals, total active days, and recent gameplay trends. The base data was obtained from Kaggle-hosted player activity datasets and extended using synthetic generation to simulate diverse churn scenarios, including gradual disengagement and abrupt drop-offs. This view enables the modeling of player retention dynamics and supports early identification of churn-prone users based on evolving behavioral signals.

C. Social View Dataset

The social view dataset models player interactions as a graph structure, where nodes represent players and edges denote social relationships such as team participation, repeated co-play, or in-game interactions. Initial node-level attributes were sourced from public datasets, while edge connections were partially synthesised to simulate coordinated behaviors, including collusion and cheating rings.

Graph-based features such as node degree, clustering coefficient, and neighbourhood risk propagation were derived from this structure. This dataset supports social influence analysis and enables the detection of coordinated cheating behavior through network-level patterns. Additionally, centrality measures capture player influence within the network, while connectivity patterns help identify hubs that may facilitate the spread of cheating behavior.

D. Image-Based Dataset

The image-based dataset consists of gameplay screenshots collected from the popular multiplayer game Free Fire, representing both cheat and non-cheat scenarios. Non-cheat images include standard gameplay visuals, while cheat samples contain visual indicators such as aim-assist overlays, unauthorized UI elements, and abnormal visual cues. The dataset was manually curated and labeled into cheat and non-cheat classes to support supervised deep learning. These images are used to train a convolutional neural network for visual cheating detection, with Grad-CAM employed to generate pixel-level explanations highlighting regions influencing model decisions.

VI. FEATURE ENGINEERING AND PREPROCESSING

Effective feature engineering and preprocessing are critical for ensuring reliable predictions and meaningful explanations in the proposed multi-view framework. Since the system integrates heterogeneous data sources, view-specific preprocessing strategies are applied while maintaining consistency across models.

A. Portrait View Feature Engineering

The portrait view focuses on aggregated player-level characteristics derived from gameplay statistics. Firstly, these features, namely total matches played, winloss ratio, killdeath ratio, accuracy, average session duration, and activity frequency were manually checked to make sure that there were no missing or invalid entries. Variables with continuous values were scaled to the range between 0 and 1 using the minmax technique to have the value ranges the same for all players and to speed up the model convergence. The features that were highly correlated and had low variance were checked and deleted to minimize the duplication and the noise. The final feature set reflects the player behavior over a long period of time and is used as the input for the estimation of the cheating risk and getting the explanation with SHAP.

B. Behavioral View Preprocessing for Churn Prediction

From the behavioral perspective, temporal engagement features were created to represent player retention patterns. Various features were obtained from raw activity logs such as recent login frequency, inactivity gaps, rolling averages of session duration, and cumulative active days. Time, dependent features were pooled over several fixed observation windows to keep the level of consistency across players. Forward filling and default inactivity indicators were used to handle the missing behavioral records. Feature scaling was done to make the learning process more stable and the imbalance of classes in the case of churn labels was dealt with by controlled sampling techniques. These features that have been engineered make it possible to predict churn with a high degree of accuracy and at the same time provide interpretable explanations.

C. Social View Feature Construction

The social view represents player interactions as a graph structure. Players are modeled as nodes, and edges represent repeated co-play or interaction relationships. From this graph, node-level features such as degree centrality, neighborhood size, and interaction frequency were extracted. To capture social risk propagation, neighborhood-based statistics were computed, reflecting exposure to suspicious or churn-prone players. These features were normalized and aligned with portrait and behavioral identifiers, enabling joint analysis across views. Social features support both risk detection and graph-based visualization for explainability.

D. Image View Preprocessing

Gameplay images used for visual cheating detection were resized to a fixed resolution and converted to a consistent color format. Pixel values were normalized to improve numerical stability during training. Data augmentation techniques such as horizontal flipping and brightness adjustment were applied to improve model generalization. Images were labeled into cheat and non-cheat classes based on visual indicators. These preprocessing steps ensure robust CNN training and enable reliable Grad-CAM visualization of regions influencing model predictions.

E. Alignment for Explainability

To support explainable AI, all engineered features were carefully mapped to human-interpretable attributes. Feature names were preserved throughout the pipeline to enable direct SHAP attribution at the player level. View-wise preprocessing consistency ensures that explanations generated across portrait, behavioral, social, and image views remain coherent and comparable. This design enables transparent interpretation of model decisions for moderators, players, and system analysts, while supporting informed decision-making through clear, human-interpretable explanations and visual evidence.

VII. MODEL DEVELOPMENT AND TRAINING

This section describes the machine learning and deep learning models used in each analytical view of the proposed framework, along with the training procedures and hyperparameter settings. The objective is to achieve accurate predictions while maintaining compatibility with explainable AI techniques.

A. Portrait View Models for Cheating and Churn Prediction

The portrait view uses aggregated player-level features to perform binary classification for cheating detection and churn prediction. Random Forest classifiers were selected due to their robustness to noise, ability to model non-linear relationships, and compatibility with feature-level explainability using SHAP.

Given a feature vector $\mathrm{x} = [x_1, x_2, \ldots, x_n]$, a Random Forest predicts the class label by aggregating the outputs of multiple decision trees:

$$\hat{y} = \mathrm{mode}\{T_1(\mathrm{x}), T_2(\mathrm{x}), \ldots, T_M(\mathrm{x})\}$$

where $T_i$denotes the i-th decision tree and $M$is the number of trees.

Two separate Random Forest models were trained:

- Cheating Detection Model
- Churn Prediction Model

Training setup:

- Train–test split: 80% / 20%
- Stratified sampling to preserve class distribution
- Evaluation metrics: accuracy and class-wise precision–recall

Hyperparameters:

- Number of trees ($n\_estimators$): 300
- Minimum samples per leaf: 5
- Random state: 42
- Parallel training enabled

These settings provided stable performance while preventing overfitting. Both models were saved and later used for SHAP-based explainability.

B. Behavioral View Model for Churn Prediction

The behavioral view focuses on temporal engagement patterns to predict player churn. A Random Forest classifier was employed due to its ability to handle heterogeneous behavioral features and non-linear relationships.

Behavioral features include login frequency, session duration, social interactions, progression indicators, and a discretized inactivity gap feature. The inactivity gap was transformed into ordinal buckets to capture disengagement trends more effectively.

Training setup:

- Train–test split: 70% / 30%
- Stratified sampling
- Evaluation metric: classification report and accuracy

Hyperparameters:

- Number of trees: 200
- Maximum tree depth: 7
- Random state: 42

This model outputs a churn probability score, which is later explained using SHAP to identify key behavioral drivers of disengagement.

C. Social View Model for Risk Propagation

The social view models player interactions as a graph and performs social risk classification. Node-level graph features such as degree centrality, betweenness, closeness, PageRank score, and cheater-neighbor statistics were extracted and used as input to a Random Forest classifier.

A synthetic social risk label was defined based on exposure to suspicious neighbors and network centrality, enabling supervised training.

Training setup:

- Train–test split: 75% / 25%
- Stratified sampling
- Class balancing enabled

Hyperparameters:

- Number of trees: 200
- Maximum depth: 8
- Class weight: balanced
- Random state: 42

This model captures coordinated cheating behavior and social exposure risks, and its predictions are supported by both SHAP explanations and graph visualizations.

D. Image View Model for Visual Cheating Detection

The image view uses a Convolutional Neural Network (CNN) to detect visual cheating cues from gameplay screenshots, such as ESP overlays and abnormal UI elements. Images were resized and normalized before training.

The CNN learns hierarchical feature representations through convolutional and pooling layers, followed by fully connected layers for classification. While churn can be estimated from portrait features, the behavioral view serves as the primary temporal churn predictor.

The output represents the probability of cheating:

$$P(\text{cheat} \mid \text{I}) = \sigma(f(\text{I}))$$

where I is the input image and $\sigma$is the sigmoid activation.

CNN architecture:

- Convolution layers: 32 → 64 → 128 filters
- Kernel size: 3×3
- Max-pooling after each convolution
- Fully connected layer with 128 units
- Dropout rate: 0.5
- Output layer: sigmoid activation

Training setup:

- Optimizer: Adam
- Learning rate: 1×10^{-4}
- Loss function: binary cross-entropy
- Batch size: 32
- Epochs: 20
- Class weighting applied to handle imbalance

The trained model produces pixel-level explanations using Grad-CAM to highlight regions influencing the cheating prediction.

E. Explainability-Aware Training Design

All models were trained with explainability in mind. Tree-based models enable SHAP-based feature attribution, while the CNN architecture supports Grad-CAM visualization. Feature consistency was preserved between training and inference to ensure faithful explanations. This design ensures that every prediction produced by the system can be traced back to meaningful behavioral, social, or visual evidence.

VIII. EXPLAINABILITY AND XAI TECHNIQUES

Explainability is a core design principle of the proposed multi-view framework, ensuring that every cheating detection and churn prediction decision can be understood, validated, and reviewed by human stakeholders. To achieve this, view-specific explainable artificial intelligence (XAI) techniques are integrated across portrait, behavioral, social, and image-based models. These explanations support transparency, fairness, and human-in-the-loop moderation.

A. SHAP-Based Explainability for Tabular Models

For the portrait, behavioral, and social views, tree-based Random Forest classifiers are used, enabling feature-level explainability through SHAP (SHapley

Additive exPlanations). SHAP attributes a prediction to individual input features based on cooperative game theory, ensuring consistent and locally accurate explanations.

Given a model fand an input feature vector x, the prediction is expressed as:

$$f(x) = \phi_0 + \sum_{i=1}^{n} \phi_i$$

where $\phi_0$represents the base value (expected model output), and $\phi_i$denotes the contribution of feature itoward the final prediction.

In the proposed system:

- Global SHAP explanations are generated to identify the most influential features across the dataset, helping moderators and analysts understand overall model behavior.
- Local SHAP explanations are produced for individual players to explain why a specific player was flagged as a cheater or predicted to churn.

By selecting the positive class SHAP values, the explanations focus explicitly on factors contributing to cheating risk or churn likelihood.

These explanations enable systematic auditing of model decisions by linking predictions to concrete feature contributions. View-wise SHAP analysis allows comparison of risk factors across portrait, behavioral, and social perspectives.

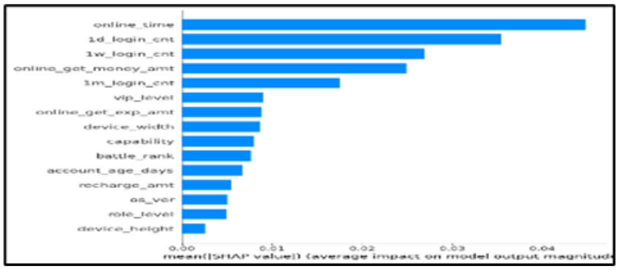

Fig.3.1 Global SHAP Feature Importance for Cheat Prediction

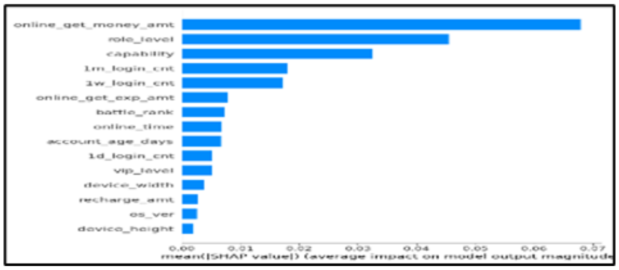

Fig.3.2 Global SHAP Feature Importance for Churn Prediction

B. Explainability for Portrait and Behavioral Views

In the portrait view, SHAP explanations highlight long-term player attributes such as activity frequency, account age, progression metrics, and device-related features that influence cheating and churn predictions. These explanations allow moderators to verify whether a Flagged decision aligns with reasonable gameplay behavior.

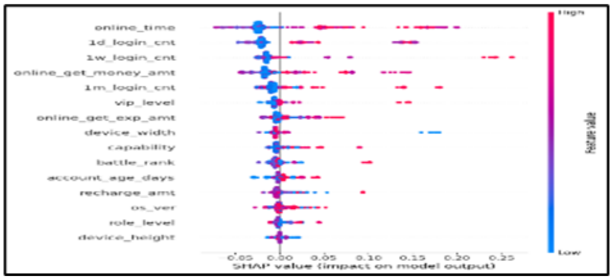

Fig.4.1. Global SHAP summary for cheating detection

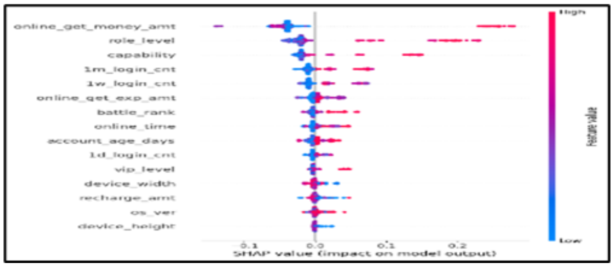

Fig.4.2. Global SHAP summary for churn detection.

In the behavioral view, SHAP is Used to explain churn predictions based on engagement dynamics, including login gaps, session duration, social activity, and recent progression. This enables developers and analysts to identify actionable churn drivers and supports transparent retention strategies.

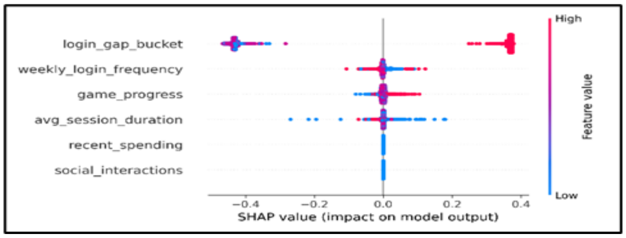

Fig.5.1. Global SHAP Summary Behavior for Churn Prediction

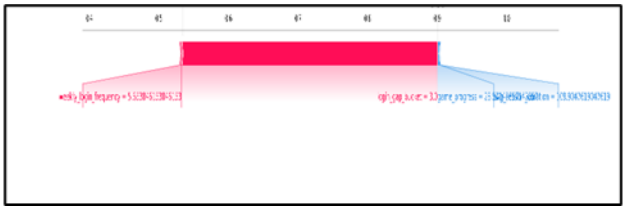

Fig.5.2. Local SHAP Explanation Behavior for Churn Prediction

C. Social View Interpretability

The social view focuses on explaining risk propagation through player interaction networks. In addition to SHAP-based feature attribution on graph-derived features, the system provides network-level visualization to interpret social influence.

Explanations include:

- The number of suspicious or cheating neighbors
- The proportion of risky connections
- Centrality-based influence indicators These explanations help moderators understand whether a player is flagged due to individual behavior or exposure within a risky social cluster, reducing the likelihood of unfair penalties.

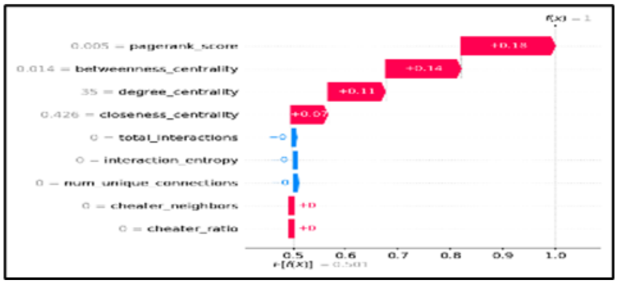

Fig.6.1. Local SHAP explanation for social view risk prediction

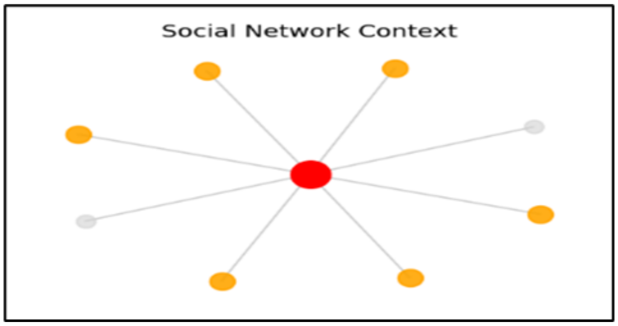

Fig. 6.2. Social network visualization for player risk analysis.

D. Grad-CAM for Visual Cheating Detection

For the image-based cheating detection model, Gradient-weighted Class Activation Mapping (Grad-CAM) is employed to provide visual explanations of CNN predictions. Grad-CAM highlights image regions that contribute most strongly to the predicted cheating probability. Formally, Grad-CAM computes a weighted combination of feature maps using gradients of the target class score:

$$L^{c}_{\text{Grad-CAM}} = \text{ReLU}\left(\sum_{k} \alpha^{c}_{k} A^{k}\right)$$

where A^krepresents the k-th convolutional feature map and $\alpha^{c}_{k}$is the importance weight derived from gradients.

In the proposed framework, Grad-CAM heatmaps reveal visual indicators such as ESP overlays, abnormal UI elements, or suspicious visual cues, allowing moderators and players to visually verify the basis of a cheating decision.

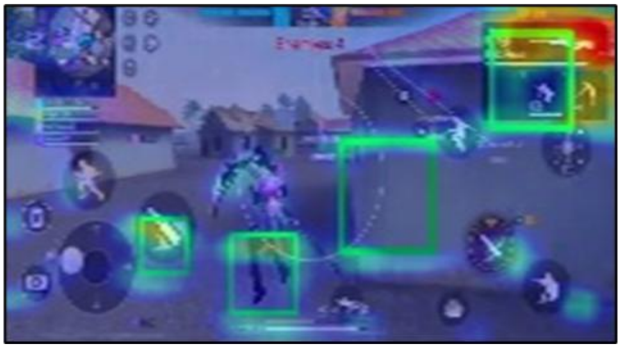

Fig.7 Grad-CAM Visualization for Image-Based Cheating Detection

E. Human-in-the-Loop Feedback Integration

Beyond passive explanation, the system incorporates a feedback mechanism for decisions generated by each analytical view. For every flagged outcome, moderators and players can provide feedback on the correctness of the decision. This feedback supports:

- Validation of automated predictions
- Identification of systematic model errors
- Continuous improvement and future retraining

By combining explainability with feedback, the system supports responsible AI deployment and strengthens trust between players, developers, and automated decision-making systems.

F. Benefits of Multi-View Explainability

The integration of SHAP, social network interpretation, and Grad-CAM ensures that explanations are view-specific, consistent, and human-interpretable. Rather than offering a single opaque risk score, the proposed framework explains why a decision was made from behavioral, social, and visual perspectives. This multi-view explainability significantly reduces false accusations, improves auditability, and enables ethical moderation in real-world online gaming environments.

IX. RESULTS AND EVALUATION

This section evaluates the performance of the proposed explainable multi-view framework for cheating detection and player churn prediction. All results are reported on held-out test datasets to assess generalization performance. Evaluation focuses on both predictive accuracy and interpretability to ensure practical usability.

A. Experimental Setup

Each model was evaluated using a train–test split as described in the model development section. Performance was measured using standard classification metrics, including accuracy, precision, recall, and F1-score. These metrics provide a balanced evaluation, particularly for imbalanced class distributions commonly observed in cheating and churn datasets.

B. Portrait View Results

The portrait- view Random Forest models were evaluated separately for cheating detection and churn prediction. Both models achieved strong classification performance on unseen test data, demonstrating the effectiveness of aggregated player-level features.

A confusion matrix was analysed to examine misclassification patterns, as shown in Fig. 8. While the majority of cheaters and non-cheaters were correctly identified, a small number of false positives were observed, highlighting the importance of explainable decision-making before enforcement actions. To support interpretability, global SHAP summary plots (Fig. 4.1 and Fig. 4.2) were generated for the cheating and churn models, respectively, identifying the most influential features contributing to model predictions. These results confirm that the portrait view models rely on meaningful behavioral attributes rather than spurious correlations.

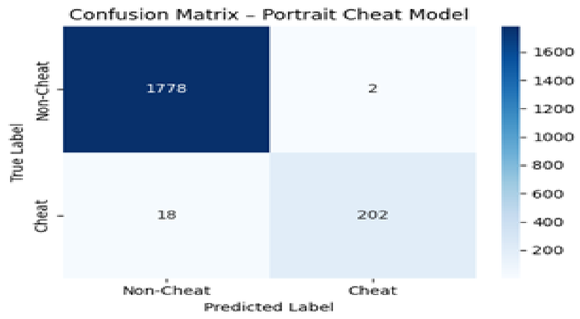

Fig.8. Confusion matrix of the portrait-view cheat detection model.

C. Behavioral View Results

The behavioral view churn prediction model demonstrated reliable performance in identifying disengagement risk based on temporal engagement patterns. Precision–recall analysis showed balanced performance across churn and non-churn classes.

SHAP-based local explanations (Fig. 5.2) were used to analyze individual churn predictions, revealing that inactivity gaps, reduced login frequency, and declining engagement were dominant contributors to churn risk. These explanations enable developers and analysts to understand and act upon churn drivers rather than relying solely on aggregate risk scores.

D. Social View Evaluation

The social view model effectively identified players with elevated social risk based on network structure and exposure to suspicious neighbors. Evaluation results indicate that social features such as cheater neighbor ratio and centrality metrics play a significant role in risk classification.

Graph-based visualizations (Fig. 6.2) and SHAP-based local explanations (Fig. 6.1) further support qualitative evaluation by illustrating how cheating behavior propagates within player networks. This view complements individual behavior analysis by capturing coordinated and socially influenced risks.

E. Image View Results

The CNN-based image model achieved strong performance in distinguishing cheat and non-cheat gameplay images. Validation accuracy remained stable across training epochs, indicating effective generalisation.

Grad-CAM visualisations were generated for representative samples to qualitatively evaluate model decisions. The highlighted regions correspond to known cheating indicators such as ESP overlays and abnormal UI elements, confirming that the model focuses on semantically meaningful visual cues rather than background artifacts.

F. Multi-View Decision Analysis

The CNN-based image model achieved strong performance in distinguishing cheat and non-cheat gameplay images. Validation accuracy remained stable across training epochs, indicating effective generalisation.

Grad-CAM visualisations (Fig. 7) were generated for representative samples to qualitatively evaluate model decisions. The highlighted regions correspond to known cheating indicators such as ESP overlays and abnormal UI elements, confirming that the model focuses on semantically meaningful visual cues rather than background artifacts.

REFERENCES

[1] L. V. Fernandes, C. D. Castanho, and R. P. Jacobi, “A survey on game analytics in massive multiplayer online games,” *Proc. 17th Brazilian Symp. Computer Games and Digital Entertainment (SBGames)*, 2018.

[2] J. Yan, “Security in computer games: An overview,” *IEEE Security & Privacy*, vol. 7, no. 3, pp. 37–44, May–June 2009.

[3] J. Yan and B. Randell, “A systematic classification of cheating in online games,” *Proc. ACM Future Play Conf.*, 2005.

[4] S. Mitterhofer, C. Kruegel, E. Kirda, and C. Platzer, “Detecting cheating in online games using behavioral features,” *Proc. Int. Workshop Recent Advances in Intrusion Detection*, 2009.

[5] T. T. Nguyen, Z. Shen, and Y. Han, “Unsupervised anomaly detection in online games,” *Proc. Int. Conf. Advances in Computer Entertainment*, 2015.

[6] M. Willman, “Machine learning to identify cheaters in online games,” M.S. thesis, Umeå Univ., Umeå, Sweden, 2020.

[7] J. P. Pinto, B. D. Schultz, and A. M. Rocha, “Deep learning and multivariate time series for cheat detection in online games,” *Neural Computing and Applications*, 2021.

[8] F. Hussein, R. Baker, and S. Alshamrani, “Automatic cheating detection using machine learning,” *Data*, vol. 7, no. 12, pp. 1–17, 2022.

[9] S. Chen, J. Chen, and H. Lin, “Game bot detection via user behavior analysis,” *Expert Systems with Applications*, vol. 57, pp. 364–372, 2016.

[10] J. Blackburn, N. Kourtellis, J. Skvoretz, M. Ripeanu, and A. Iamnitchi, “Cheating in online games: A social network perspective,” *ACM Trans. Internet Technology*, vol. 13, no. 3, 2014.

[11] Y.-S. Shih, T.-Y. Li, and C.-W. Wang, “Human-in-the-loop AI for cheating ring detection,” *IEEE Access*, vol. 12, pp. 113400–113412, 2024.

[12] K. Mustač, G. Pranjić, and G. Mekterović, “Predicting player churn in free-to-play mobile games,” *Applied Sciences*, vol. 12, no. 15, 2022.

[13] E. Loria, S. Rossi, and G. Marfia, “Exploiting limited player behavioral data to predict churn,” *Entertainment Computing*, vol. 39, 2021.

[14] J. Runge, P. Gao, F. Garcin, and B. Faltings, “Churn prediction for high-value players in social games,” *Proc. IEEE Conf. Computational Intelligence and Games*, pp. 1–8, 2014.

[15] M. Hadiji, “Predicting churn in online games using deep learning,” *Proc. AAAI Workshops*, 2014.

[16] Y.-J. Han *et al.*, “Prediction of churning game users with social activity and graph neural networks,” *IEEE Transactions on Games*, 2024.

[17] J. Tao, Y. Xiong, S. Zhao, R. Wu, X. Shen, T. Lyu, C. Fan, Z. Hu, S. Zhao, and G. Pan, “Explainable AI for cheating detection and churn prediction in online games,” *IEEE Transactions on Games*, vol. 15, no. 2, pp. 242–251, 2023.

[18] J. Tao *et al.*, “XAI-driven explainable multi-view game cheating detection,” *Proc. IEEE Conf. on Games*, pp. 144–151, 2020.

[19] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[20] S. M. Lundberg *et al.*, “Consistent individualised feature attribution for tree ensembles,” *Proc. Machine Learning Research*, 2017.

[21] R. R. Selvaraju *et al.*, “Grad-CAM: Visual explanations from deep networks via gradient-based localisation,” *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, pp. 618–626, 2017.

[22] D. Y. C. Wang, L. W. Shiu, and Y. C. Lin, “Explainability of highly associated fuzzy churn patterns in binary classification,” *IEEE Access*, vol. 12, pp. 45522–45536, 2024.

[23] J. Maan and H. Maan, “Customer churn prediction model using explainable machine learning,” *Int. J. Information Technology*, vol. 15, 2023.

[24] R. K. Gupta and A. Srivastava, “Explicit and implicit features for interpretable churn prediction in online games,” *ACM Trans. Interactive Intelligent Systems*, vol. 13, no. 3, 2023.

[25] J. Herbrich, T. Minka, and T. Graepel, “TrueSkill™: A Bayesian skill rating system,” Microsoft Research Tech. Rep., 2007.