THIẾT KẾ PIPELINE REAL-TIME PLAYER ENGAGEMENT VÀ ANALYTICS CHO CÁC NỀN TẢNG GAME HIỆN ĐẠI

Mai Phan Anh Tùng, Phạm Hải Tiến, Phạm Quốc Hùng

Đại học Công nghệ – Đại học Quốc gia Hà Nội (VNU–UET)

Ngày 23 tháng 11 năm 2025

Tóm tắt nội dung

Báo cáo này trình bày quá trình thiết kế và xây dựng một hệ thống phân tích mức độ tương tác người chơi trong game theo thời gian thực (Real-Time Player Engagement Analytics), dựa trên sự kết hợp của các công nghệ streaming hiện đại bao gồm Apache Kafka, Spark Structured Streaming, Redis, mô hình học máy (Machine Learning) và một hệ thống dashboard trực quan hoá được xây dựng bằng Flask.

Hệ thống được thiết kế nhằm mô phỏng môi trường hoạt động của các nền tảng game trực tuyến, nơi dữ liệu hành vi người chơi liên tục phát sinh, bao gồm các sự kiện như đăng ký tài khoản (register), đăng nhập (login), đăng xuất (logout), mua vật phẩm (purchase) và lên cấp (level_up). Các sự kiện này được gửi vào Kafka theo thời gian thực bởi một mô-đun mô phỏng (game event simulator). Spark Structured Streaming chịu trách nhiệm đọc luồng dữ liệu này theo micro-batch, cập nhật trạng thái hành vi người chơi trong Redis, tính toán lại các chỉ số phái sinh và kích hoạt mô hình học máy để dự đoán mức độ tương tác (Engagement Level).

Bằng cách kết hợp xử lý thời gian thực, mô hình học máy và kho dữ liệu in-memory, hệ thống có thể theo dõi trạng thái người chơi, đưa ra dự đoán tức thời và hỗ trợ các quyết định cá nhân hoá như điều chỉnh độ khó, gợi ý phần thưởng hoặc cảnh báo hành vi bất thường. Mặc dù dữ liệu được mô phỏng, pipeline này phản ánh cấu trúc thực tế của các hệ thống phân tích hành vi người chơi được sử dụng trong nhiều nền tảng game hiện đại.

Mục lục

1 Giới Thiệu **3**
1.1 Hạn Chế Thu Thập Dữ Liệu Real-Time và Giải Pháp Mô Phỏng 3

2 Giới Thiệu Các Công Nghệ Sử Dụng **4**
2.1 Apache Kafka 4
2.2 Apache Spark Structured Streaming 5
2.3 Redis 5
2.4 PySpark MLlib (Machine Learning) 5
2.5 Flask 5
2.6 Docker 6

3 Phân Tích Dữ Liệu Offline (Offline Data Analytics) **6**
3.1 So Sánh Hành Vi Giữa Các Nhóm EngagementLevel 6
3.2 Ma Trận Tương Quan 7
3.3 Phân Cụm Người Chơi (Clustering) 7
3.4 Kết Luận Phần Phân Tích Dữ Liệu 7

4 Xây dựng mô hình bằng PySpark ML **7**
4.1 Tiền xử lý và Feature Engineering 8
4.2 Xây dựng Pipeline huấn luyện 8
4.3 Tối ưu siêu tham số bằng Cross-Validation 8
4.4 Kết quả huấn luyện 9

5 Pipeline Xử Lý Dữ Liệu Real-Time **9**
5.1 Mô phỏng sự kiện người chơi (Kafka Producer) 9
5.2 Truyền tải sự kiện (Kafka Broker) 9
5.3 Xử lý streaming (Spark Structured Streaming) 9
5.4 Lưu trữ real-time và dự đoán (Redis + ML Model) 10
5.5 Trực quan hoá và API (Flask Dashboard) 10

6 Kết Luận và Hướng Phát Triển **10**

1 Giới Thiệu

Trong những năm gần đây, ngành công nghiệp game trực tuyến phát triển mạnh mẽ với hàng trăm triệu người chơi hoạt động hàng ngày trên các nền tảng như Steam, PlayStation Network, Xbox Live và các hệ thống game di động. Khối lượng dữ liệu sinh ra từ hành vi người chơi, từ các tương tác trong game đến các giao dịch mua vật phẩm, tạo nên một nguồn dữ liệu khổng lồ và liên tục theo thời gian.

Để tối ưu trải nghiệm người dùng và tăng khả năng giữ chân (retention), các nền tảng game hiện đại đều ứng dụng các hệ thống phân tích thời gian thực nhằm xử lý lượng lớn sự kiện phát sinh liên tục. Bên cạnh đó, vị thế cạnh tranh trong thị trường game đòi hỏi các nhà phát triển phải hiểu sâu sắc hành vi người chơi để cá nhân hóa nội dung, phát hiện bất thường, điều chỉnh độ khó động (dynamic difficulty) và vận hành các chiến dịch marketing theo thời gian thực.

Một trong những mục tiêu quan trọng nhất của phân tích real-time trong ngành game là **dự đoán mức độ tương tác của người chơi (player engagement prediction)**. Người chơi có mức tương tác cao thường:

- có thời gian chơi dài hơn,
- hoàn thành nhiều nhiệm vụ hơn,
- mua nhiều vật phẩm hơn,
- và có xu hướng gắn bó lâu dài với sản phẩm.

Do đó, việc theo dõi và dự đoán tương tác theo thời gian thực có thể giúp nhà vận hành game:

- tùy chỉnh độ khó theo từng người chơi,
- gợi ý phần thưởng chính xác theo hành vi,
- phát hiện người chơi có nguy cơ rời bỏ game,
- tối ưu hóa hệ thống gợi ý nhiệm vụ,
- điều phối các chiến dịch marketing cá nhân hóa,
- và cân bằng tải hệ thống dựa trên hành vi người chơi.

Ngoài ra, dữ liệu game sở hữu đầy đủ ba đặc trưng của Big Data: *Volume*, *Velocity*, và *Variety*. Điều này làm cho việc thiết kế một pipeline xử lý dữ liệu real-time trở nên thiết yếu để đảm bảo khả năng mở rộng, độ trễ thấp và độ tin cậy cao.

Mặc dù nhu cầu phân tích hành vi real-time là rất lớn, việc thu thập trực tiếp dữ liệu telemetry của game thực tế thường gặp nhiều hạn chế.

1.1 Hạn Chế Thu Thập Dữ Liệu Real-Time và Giải Pháp Mô Phỏng

Một trong những thách thức lớn nhất khi xây dựng hệ thống phân tích hành vi người chơi theo thời gian thực là việc không thể tiếp cận trực tiếp nguồn dữ liệu streaming từ các nền tảng game thương mại. Hầu hết các trò chơi trực tuyến phổ biến đều không cung cấp API công khai cho phép truy cập dữ liệu telemetry real-time do các yếu tố liên quan đến:

- bảo mật và quyền riêng tư người dùng,

- hạn chế truy cập hạ tầng backend,
- rủi ro tải hệ thống nếu mở API real-time,
- chính sách bản quyền và khai thác dữ liệu,
- nguồn dữ liệu huấn luyện bắt buộc phải lấy từ kho dữ liệu công khai thay vì hệ thống sản xuất.

Vì vậy, dữ liệu thời gian thực không thể thu thập trực tiếp từ môi trường game thực tế trong phạm vi nghiên cứu này. Do đó, nhóm sử dụng bộ dữ liệu công khai từ Kaggle:

- **Predict Online Gaming Behavior Dataset** – Kaggle
  https://www.kaggle.com/datasets/rabieelkharoua/predict-online-gaming-behavior-da

Bộ dữ liệu này chứa các thông tin mô tả hành vi và hồ sơ người chơi như tần suất chơi, thời lượng trung bình mỗi phiên, cấp độ nhân vật, số lượng thành tựu và hành vi chi tiêu trong game. Tuy nhiên, đây là dữ liệu dạng *batch*, không phải dữ liệu thời gian thực (streaming).

Để đáp ứng yêu cầu của pipeline real-time, dữ liệu được chuyển đổi thành dạng sự kiện (event stream) bằng một mô-đun mô phỏng hành vi người chơi (Kafka Producer):

- mỗi dòng dữ liệu hồ sơ người chơi được nạp vào hệ thống như trạng thái khởi tạo,
- các sự kiện như login/logout/purchase/level_up được sinh ngẫu nhiên theo phân phối hợp lý,
- dữ liệu mô phỏng được đẩy liên tục vào Kafka tạo thành luồng sự kiện gần giống với hành vi của người chơi thật,
- Spark Structured Streaming xử lý các sự kiện này như thể chúng là telemetry thực tế.

Cách tiếp cận này đảm bảo rằng pipeline có thể được thử nghiệm, đánh giá và tinh chỉnh trong điều kiện gần giống môi trường game sản xuất, mặc dù dữ liệu gốc không đến từ API real-time. Quan trọng hơn, kiến trúc pipeline được thiết kế sao cho nó có thể dễ dàng được tích hợp với dữ liệu thực khi nguồn telemetry real-time khả dụng trong tương lai.

2 Giới Thiệu Các Công Nghệ Sử Dụng

Hệ thống được xây dựng dựa trên nhiều công nghệ xử lý dữ liệu hiện đại, mỗi thành phần đảm nhiệm một vai trò khác nhau trong pipeline real-time. Các công nghệ chính bao gồm:

2.1 Apache Kafka

Apache Kafka là nền tảng streaming phân tán chuyên dụng để xử lý dữ liệu theo mô hình publish–subscribe. Kafka có khả năng xử lý hàng triệu sự kiện mỗi giây với độ trễ thấp, độ bền dữ liệu cao và được sử dụng rộng rãi trong các hệ thống game, tài chính, và IoT. Trong nghiên cứu này, Kafka đóng vai trò:

- nhận luồng sự kiện người chơi mô phỏng từ Kafka Producer,
- lưu trữ tạm thời và phân phối sự kiện cho Spark Structured Streaming.

2.2 Apache Spark Structured Streaming

Structured Streaming là mô hình xử lý streaming của Spark cho phép xử lý dữ liệu theo micro-batch với API thống nhất với các thao tác DataFrame. Spark phù hợp cho xử lý dữ liệu tốc độ cao và tích hợp tốt với Kafka. Trong hệ thống:

- Spark đọc dữ liệu từ Kafka theo chu kỳ 2 giây,
- xử lý sự kiện, cập nhật dữ liệu người chơi,
- tính toán các chỉ số phái sinh,
- kích hoạt mô hình học máy để dự đoán mức độ tương tác.

2.3 Redis

Redis là hệ quản trị cơ sở dữ liệu in-memory có hiệu năng cực cao, hỗ trợ truy vấn và cập nhật chỉ trong vài mili-giây. Hệ thống sử dụng Redis để:

- lưu trạng thái real-time của từng người chơi,
- lưu danh sách sự kiện gần nhất,
- lưu live metrics phục vụ dashboard.

2.4 PySpark MLlib (Machine Learning)

Hệ thống sử dụng thư viện `pyspark.ml` để xây dựng mô hình dự đoán mức độ tương tác (EngagementLevel). Khác với các thư viện học máy dạng batch như Scikit-Learn, PySpark hỗ trợ xử lý phân tán và phù hợp với mô hình dữ liệu lớn (Big Data).

Mô hình chính được sử dụng là *Random Forest Classifier*, kết hợp với **Pipeline** và **CrossValidator** để tự động hóa toàn bộ quy trình:

- tiền xử lý dữ liệu,
- mã hóa các đặc trưng dạng chuỗi,
- ghép đặc trưng vào vector đầu vào,
- huấn luyện mô hình Random Forest,
- đánh giá mô hình bằng Cross-Validation và Grid Search.

Sau khi tối ưu, mô hình có độ chính xác **88.15%** và được lưu dưới dạng mô hình Spark ML tại thư mục `cv_pipeline_model`.

2.5 Flask

Flask đóng vai trò backend API phục vụ dashboard trực quan hoá. Đây là framework Python nhẹ, dễ tích hợp, phù hợp cho real-time monitoring. Flask truy vấn Redis để trả về:

- thống kê tổng quan,
- dữ liệu người chơi,
- danh sách sự kiện gần nhất.

2.6 Docker

Toàn bộ hệ thống được container hóa bằng Docker, đảm bảo các thành phần có thể triển khai đồng nhất trên mọi môi trường. Mỗi thành phần của pipeline được triển khai dưới dạng một container độc lập.

3 Phân Tích Dữ Liệu Offline (Offline Data Analytics)

Để hỗ trợ quá trình xây dựng mô hình học máy và hiểu rõ hành vi người chơi, một quy trình phân tích dữ liệu (Exploratory Data Analysis – EDA) được tiến hành trên tập dữ liệu `online_gaming_behavior_dat` Việc phân tích giúp xác định phân phối, mức độ biến động, tương quan giữa các biến và nhận diện những đặc trưng quan trọng ảnh hưởng đến mức độ tương tác (EngagementLevel). Dữ liệu được xử lý bằng Python cùng các thư viện pandas, numpy và matplotlib.

Tập dữ liệu bao gồm các nhóm đặc trưng chính:

- **Nhân khẩu học**: Age, Gender, Location.
- **Hành vi chơi**: SessionsPerWeek, AvgSessionDurationMinutes, PlayTimeHours.
- **Tiến trình**: PlayerLevel, AchievementsUnlocked.
- **Chi tiêu**: InGamePurchases.
- **Độ khó trò chơi**: GameDifficulty.
- **Nhãn mục tiêu**: EngagementLevel.

Ngoài ra, một đặc trưng mới được tạo trong quá trình tiền xử lý:

- **isAddicted**: giá trị bằng 1 khi thời lượng chơi trung bình tuần vượt quá 1280 phút, ngược lại bằng 0.

Phân tích mô tả cho thấy:

- PlayTimeHours và SessionsPerWeek có phân phối lệch phải do một nhóm nhỏ người chơi có thời gian chơi rất cao.
- AchievementsUnlocked và PlayerLevel tăng dần theo mức độ tương tác.
- Các biến phân loại như Gender, Location và GameGenre có phân phối tương đối đồng đều, tránh bias về tập người chơi.

3.1 So Sánh Hành Vi Giữa Các Nhóm EngagementLevel

Dữ liệu được nhóm theo nhãn EngagementLevel và tính trung bình các biến hành vi:

- Nhóm **High Engagement** có trung bình PlayTimeHours, AchievementsUnlocked và PlayerLevel cao hơn hai nhóm còn lại.
- Nhóm **Low Engagement** có tần suất chơi và tổng thời gian chơi thấp nhất.
- Điều này hỗ trợ mô hình phân loại với ngưỡng phân chia rõ ràng giữa các nhóm.

3.2 Ma Trận Tương Quan

Phân tích tương quan Pearson giữa các biến số cho thấy:

- PlayTimeHours tương quan mạnh với AchievementsUnlocked và PlayerLevel.
- AvgSessionDurationMinutes tương quan với SessionsPerWeek, cho thấy người chơi chơi thường xuyên cũng có xu hướng chơi lâu hơn mỗi phiên.
- Chi tiêu trong game (InGamePurchases) có tương quan đáng kể với EngagementLevel nhưng không quá mạnh, phản ánh thực tế người chơi chi tiêu không nhất thiết là người tương tác cao.

Các mối tương quan này xác nhận rằng mô hình dự đoán EngagementLevel nên dựa chủ yếu vào đặc trưng hành vi và tiến trình người chơi.

3.3 Phân Cụm Người Chơi (Clustering)

K-Means được áp dụng trên bộ đặc trưng:

- PlayTimeHours
- SessionsPerWeek
- AvgSessionDurationMinutes

Kết quả cho thấy sự hình thành rõ ràng của ba nhóm người chơi:

1. **Nhóm casual**: tần suất và thời lượng chơi thấp.
2. **Nhóm mid-core**: chơi đều, thời lượng trung bình.
3. **Nhóm hardcore**: chơi lâu, nhiều session và thời lượng cao.

Kết quả phân cụm phù hợp với ba nhóm EngagementLevel trong dataset. Điều này xác nhận rằng phân nhóm tự nhiên của dữ liệu trùng khớp với cấu trúc nhãn, giúp củng cố chất lượng mô hình phân loại.

3.4 Kết Luận Phần Phân Tích Dữ Liệu

Phân tích EDA cho thấy các đặc trưng hành vi như SessionsPerWeek, AvgSessionDurationMinutes và PlayerLevel đóng vai trò quan trọng trong việc phân biệt mức độ tương tác. Ma trận tương quan và phân tích clustering củng cố thêm nhận định này. Kết quả phân tích giúp tối ưu hoá pipeline machine learning và chứng minh rằng dữ liệu mô phỏng có cấu trúc hợp lý, phù hợp để đưa vào hệ thống real-time.

4 Xây dựng mô hình bằng PySpark ML

Quy trình huấn luyện mô hình được triển khai hoàn toàn bằng `pyspark.ml`. Toàn bộ tiến trình bao gồm tiền xử lý dữ liệu, tạo pipeline, tối ưu siêu tham số và đánh giá mô hình.

4.1 Tiền xử lý và Feature Engineering

Các cột được sử dụng gồm:

- Age, Gender, Location, GameGenre,
- InGamePurchases, SessionsPerWeek, AvgSessionDurationMinutes,
- PlayerLevel, AchievementsUnlocked, GameDifficulty, EngagementLevel.

Các bước tiền xử lý:

- Chuẩn hóa các cột chuỗi bằng **StringIndexer**.
- Chuyển thành One-Hot Vector bằng **OneHotEncoder**.
- Ghép toàn bộ đặc trưng vào một vector duy nhất bằng **VectorAssembler**.
- Tạo thêm cột **isAddicted** theo quy tắc:

$$\text{isAddicted} = \begin{cases} 1, & \text{AvgSessionDurationMinutes} > 1280 \\ 0, & \text{otherwise} \end{cases}$$

4.2 Xây dựng Pipeline huấn luyện

Pipeline được xây dựng như sau:

```
pipeline = Pipeline(stages = indexers +
                    [encoder, label_indexer, assembler, rf])
```

Pipeline bao gồm:

- Chuỗi các bước xử lý dữ liệu (indexers),
- Bộ mã hóa One-Hot,
- Mã hóa nhãn mục tiêu,
- VectorAssembler,
- Mô hình RandomForestClassifier.

4.3 Tối ưu siêu tham số bằng Cross-Validation

Mạng lưới siêu tham số (Grid Search):

- Số cây trong rừng: numTrees = [100, 200]
- Độ sâu tối đa của cây: maxDepth = [8, 10]
- Chiến lược chọn đặc trưng: featureSubsetStrategy = ”sqrt”

Cross-Validation:

- 3-fold cross-validation
- Đánh giá bằng thước đo Accuracy
- Số luồng chạy song song: 4

4.4 Kết quả huấn luyện

- Thời gian huấn luyện: **khoảng 5 phút**.
- Độ chính xác tối ưu: **0.8815**.
- Mô hình cuối cùng được lưu tại: `cv_pipeline_model`.

5 Pipeline Xử Lý Dữ Liệu Real-Time

Hệ thống được thiết kế theo kiến trúc pipeline streaming nhằm xử lý dữ liệu người chơi theo thời gian thực từ lúc sự kiện được sinh ra cho đến lúc hiển thị trên dashboard. Pipeline bao gồm bốn lớp chức năng chính: **(1) mô phỏng dữ liệu**, **(2) truyền tải sự kiện**, **(3) xử lý streaming**, **(4) lưu trữ và dự đoán**, và **(5) trực quan hoá**. Toàn bộ thành phần vận hành liên tục và đồng bộ như trong một hệ thống game thực tế.

5.1 Mô phỏng sự kiện người chơi (Kafka Producer)

Vì không thể thu thập dữ liệu real-time từ game do hạn chế API public, dữ liệu Kaggle được sử dụng làm nguồn offline và được chuyển hóa thành luồng sự kiện real-time. Kafka Producer:

- đọc dữ liệu từ file CSV,
- tạo sự kiện *register*, *login*, *logout*, *purchase*, *level_up*,
- gửi sự kiện vào Kafka topic `game_events` theo thời gian thực.

Điều này mô phỏng hành vi người chơi tương tự như telemetry của game thật.

5.2 Truyền tải sự kiện (Kafka Broker)

Kafka chịu trách nhiệm phân phối dữ liệu giữa Producer và Spark:

- đảm bảo lưu trữ tạm thời có độ tin cậy cao,
- hỗ trợ scale-out khi có nhiều consumers,
- duy trì độ trễ thấp trong truyền tải.

5.3 Xử lý streaming (Spark Structured Streaming)

Spark là thành phần cốt lõi trong pipeline real-time:

1. đọc sự kiện từ Kafka theo micro-batch 2 giây,
2. parse JSON theo schema định nghĩa sẵn,
3. cập nhật các chỉ số thô vào Redis (TotalSessions, TotalSpent, PlayerLevel),
4. tính toán chỉ số phái sinh như AvgSessionDuration,
5. lấy dữ liệu mới nhất và chạy mô hình học máy dự đoán EngagementLevel,
6. ghi lại kết quả vào Redis.

Spark giữ vai trò “bộ não” của hệ thống, đảm bảo mọi dữ liệu đều được cập nhật liên tục.

5.4 Lưu trữ real-time và dự đoán (Redis + ML Model)

Redis đóng vai trò như kho dữ liệu trạng thái của người chơi:

- lưu trạng thái player dưới dạng Hash,
- lưu danh sách sự kiện gần nhất dưới dạng List,
- lưu các chỉ số tổng quan (live metrics),
- phản hồi dữ liệu gần như tức thời cho dashboard.

Mô hình Random Forest (được huấn luyện offline) dự đoán mức tương tác mỗi khi trạng thái người chơi thay đổi. Kết quả được ghi đè trực tiếp vào Redis.

5.5 Trực quan hoá và API (Flask Dashboard)

Dashboard được xây dựng bằng Flask + HTML/JS:

- API truy vấn Redis theo thời gian thực,
- hiển thị số người chơi online, tổng sự kiện, biểu đồ và chi tiết từng người chơi,
- cập nhật dữ liệu theo chu kỳ ngắn mà không ảnh hưởng hiệu năng hệ thống.

Pipeline đảm bảo tất cả thành phần hoạt động đồng bộ, từ mô phỏng dữ liệu đến hiển thị trên giao diện.

6 Kết Luận và Hướng Phát Triển

Hệ thống được xây dựng đã mô phỏng đầy đủ một nền tảng phân tích dữ liệu người chơi real-time, bao gồm ingestion, streaming processing, fast storage, machine learning và trực quan hoá.

Trong tương lai, có thể mở rộng:

- thêm anomaly detection,
- thêm phân tích hành vi nâng cao,
- thêm A/B testing real-time,
- tích hợp gợi ý nhiệm vụ dựa trên Reinforcement Learning.

Tài Liệu Tham Khảo

1. Apache Kafka Documentation. Available: https://kafka.apache.org/documentation/
2. Apache Spark Structured Streaming Guide. Available: https://spark.apache.org/docs/latest/structured-streaming-programming-guide.html
3. Redis In-Memory Database Documentation. Available: https://redis.io/documentation
4. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, 2011.

5. Akiba et al., “Optuna: A Next-generation Hyperparameter Optimization Framework,” KDD 2019.

6. Géron, A., “Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow,” O’Reilly Media, 2019.

7. Chen et al., “Building Real-Time Analytics Systems for Online Gaming Platforms,” IEEE Transactions on Games, 2020.

8. Kumar et al., “User Engagement Prediction in Games using Behavioral Telemetry,” ACM CHI Play, 2018.