

## Article

# Transformer-Based Behavioral Representation Learning for Gamer Engagement Detection from User Interaction Patterns

Wad Ghaban 

Department of Computer Science, Faculty of Computer Sciences and information Technology,  
University of Tabuk, Tabuk 71491, Saudi Arabia; wghaban@ut.edu.sa

## Abstract

Automated gamer engagement detection enables the detection and classification of the level of user involvement in a gaming environment, which remains a challenging issue because of high dimensionality, heterogeneity, and complex interdependencies of behavioral data. Conventional machine learning and deep learning models are mainly based on handcrafted features with limited ability to model complex nonlinear interactions among heterogeneous behavioral attributes. To address these issues, this study proposes a Transformer-based Behavior Analysis Framework for Gamer Engagement Detection (TBAF-GED) model. The proposed model first uses correlation-based feature selection to remove redundant behavioral attributes. An autoencoder is then applied to learn nonlinear representations from the selected features. Finally, a Feature Tokenizer Transformer converts the selected attributes into feature-level tokens and uses self-attention to capture their interactions. Moreover, the Nadam optimizer is applied to update model parameters and improve convergence stability during training. Experimental evaluation demonstrates that the proposed TBAF-GED framework achieves superior performance, with an accuracy of 94.26% compared with existing approaches. The final outcome categorizes gamer engagement levels based on game difficulty and genre as Low, Medium, and High. The framework provides an interpretable approach to analyze engagement patterns using the available gaming behavioral attributes. The proposed model supports game analytics by enabling proactive identification of gamer engagement levels from behavioral data. It helps to improve game design decisions and enhances player retention through intelligent engagement analysis.

**Keywords:** gamer engagement detection; behavior analysis; Feature Tokenizer Transformer; deep learning; virtual gaming

## 1. Introduction

In recent years, online games have progressed into a complex digital ecosystem that facilitates players' entertainment, social interaction, and economic activities [1]. In these digital environments, player engagement plays a key role in long-term participation, game progression, and monetization outcomes. The continuous engagement of players not only supports the longevity of online games but also influences the decisions associated with content updates [2]. Therefore, analyzing the factors that affect player engagement is an essential focus in game analytics and behavioral data science. Player engagement is a multi-dimensional concept that includes gameplay involvement, playing duration, and progression within the game [3]. Earlier researchers have analyzed player engagement based on various demographic information and gameplay-related features such as age, gender, region, genre, playing duration, level of difficulty, in-game spending, achievements,



Received: 13 August 2026

Revised: 30 September 2026

Accepted: 6 October 2026

Published: 10 October 2026

**Copyright:** © 2026 by the author.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

and engagement activities [4]. These features are very important for designing a predictive model that can improve the user experience and player retention. The main challenge in the gaming industry is to estimate players' engagement levels [5]. Player engagement strongly influences the game's popularity and income generation. However, predicting player engagement levels is difficult due to the differences and complexity in player behavior [6].

Advances in machine learning (ML), a branch of AI, provide the ability to learn from datasets using different strategies. This allows systems to classify or forecast outcomes by simulating player behavior in advance [7]. The ML model focuses on a system that gathers data, recognizes patterns, and learns from various situations to make decisions. For instance, in digital games, the game behavior can be enhanced autonomously without the need for direct programming [8]. Despite its effectiveness, many traditional ML approaches struggle to capture the non-linear, complex, and high-dimensional nature of gaming data. The advancement of deep learning (DL) models is better than that of ML models when dealing with complex datasets, as it can learn patterns automatically from the data [9]. In addition, deep learning models use multiple layers that work in parallel, and they can automatically learn relevant features from the dataset. As a result, DL models offer more accurate and reliable outcomes compared to traditional approaches, making them well-suited for classifying and detecting player engagement level in digital gaming settings [10].

#### *Motivation and Research Contributions*

The rapid development of online gaming has created player involvement as an important factor for player retention and platform success. Conversely, engagement is impacted by risky and behavioral patterns that are complex to extract using standard models. Conventional strategies frequently struggle to fully utilize rich user interaction information. This creates a requirement for an intelligent data-driven technique that acquires significant patterns from behavioral data. To address these conditions, this research presents a Transformer-based Behavior Analysis Framework for Gamer Engagement Detection (TBAF-GED) using behavioral data. The TBAF-GED model mainly integrates the learning elements in a task-specific flow for resolving the redundancy, compact representation, and complicated feature-interaction characteristics of gamer behavioral data. The major research question of this study is whether player behavioral and progression-related features are efficiently employed to categorize various stages of gamer engagement. The study investigates how behavioral data, such as player level, unlocked achievements, and session frequency, influence the differentiation of Low, Medium, and High engagement levels. The major contribution does not rely on the introduction of a Transformer or optimization algorithm; rather, it lies in the task-specific integration of correlation-based feature selection, autoencoder-based representation learning, and FT-Transformer-based feature-interaction modeling in the TBAF-GED framework. The core contributions of this study are as follows.

- Introduce a correlation-based feature selection and autoencoder (CFS-AE) for effective elimination of redundant and irrelevant features while learning compact and informative feature representations. The CFS-AE model supports task-specific processing and a representation stage for structured gamer behavioral characteristics.
- Develop a Feature Tokenizer Transformer (FT-Transformer) approach to convert gamer behavior features into token embeddings and model their interactions using self-attention. It performs the classification of engagement levels reflecting player retention (High, Medium, and Low). The FT-Transformer model is integrated to define interactions amongst the compact behavioral representations instead of being claimed as a new Transformer architecture.

- Integrate a Nadam optimizer to efficiently update model parameters and improve training stability during optimization, while its efficiency is determined in integration with the CFS-AE and FT-Transformer components for gamer behavioral data.
- Validate the performance of the proposed model using the Player Engagement Analysis and Prediction dataset, and demonstrate its effectiveness using various evaluation measures. The integrated proposed model is validated to investigate the significance of its integrated components to gamer engagement classification.

The rest of the article is organized as follows: Section 2 presents existing related work on gamer engagement detection. Section 3 offers an overview of the proposed method, Section 4 offers the results and discussion, and Section 5 concludes the work with future directions.

## 2. Related Works

This section presents a concise overview of related works on gamer engagement detection, highlighting identified research gaps. Rashed et al. [11] developed a new architecture for real-time, nonintrusive, and indirect measurement of engagement in multiplayer online games that depends on the flow scheme. The study integrates graph convolutional networks (GCN) for modeling player connections with Transformer systems for temporal processing, empowering indirect measurement of game challenge and player skill, which is utilized to categorize player engagement. Ozyirmidokuz et al. [12] discussed the increasing prevalence of digital game addiction (DGA) by deploying an information scheme for identification and initial involvement, for predicting an individual's inclination for DGA. A data augmentation approach was employed to improve dataset diversity and improve the method's generalization ability. Subsequently, wide-ranging preprocessing was performed, and six ML strategies were assessed. Yin et al. [13] designed a new attention-driven neural point process system to extract the complexity and richness of customer behavioral dynamics in the current digital context of consumption. The method features a novel multiple-representation, continuous-time attention system that can reliably model dynamic connections among distinct categories of behavior across external effects.

Wang, Y. and Lee [14] proposed a methodology to enhance game suggestions and operations by accurately forecasting users' preferences by integrating the spatiotemporal activity and social ties of game players. Liang et al. [15] investigated a DL-driven user behavior and interaction mechanism optimization approach. To construct a BiLSTM-Attention behavior detection algorithm, it uses a behavior-detection model classifier of higher-dimensional operational information and fine-tunes interaction methodology and response hyperparameters in conjunction with the categorization solution for optimizing the game comments system. Han et al. [16] introduced an advanced churn prediction strategy that exploits user activity and social collaboration information from the huge, multi-person online role-playing game, *Soul and Blade*. This approach distinctively predicts people's interactions as a graph structure and improves forecasting precision by combining a correct and smooth (C&S) model with a GCN model in social network analysis.

Lauer-Schmaltz et al. [17] introduced the model and operation of an HDT approach for upper-limb stroke rehabilitation utilizing exoskeletons and critical games, regarding the ETHICA technique. The study validates HDT real-world modifications to therapy problems and exoskeleton support depends on player situations, improves the relationship between non-medical and medical stakeholders by decision-support systems and data visualizations, and increases player involvement via personalized comments. Marín-Lora et al. [18] examined a mobile Exergame developed particularly for treadmill running, using an interface with a smartwatch. The game poses natural settings where the smartwatch network illuminates the gamers' behavior, converting it into running speed and collabora-

tive activities by identifying signals in the game. The major goal is to deliver players with sustained gaming involvement tailored to the features of treadmill running. It particularly focuses on prioritizing the playful element of this game, identifying its relevant context of treadmill running.

To evaluate the success of the goals and the presented hypothesis, comparative research was conducted between a treadmill running simulator and the presented game.

Existing studies on gamer player engagement recognition mostly utilize statistical examination and conventional ML approaches for analyzing user behavior information, including actions, session time, and interaction occurrence. These techniques, including Random Forest and SVM, enhance prediction; however, they are highly dependent on feature engineering. Latest research investigates DL strategies like Transformers and LSTM to extract sequential and intricate behavioral patterns. AE methodologies are also used for better feature representation. The proposed model exists to work towards the goal of improving engagement information precision for better personalization and player retention.

### 3. Methods

This research follows a multi-faceted approach for gamer engagement detection using behavior data. The Player Engagement Analysis and Prediction dataset serves as the foundation for model development and evaluation. Figure 1 showcases the end-to-end architecture of the TBAF-GED approach. The proposed model initially pre-processes the input data through data cleaning, label encoding, outlier handling, and robust scaling to ensure its quality and usability. The proposed framework has three main components: correlation-based feature selection, autoencoder-based nonlinear representation learning, and FT-Transformer-based behavioral interaction modeling. The correlation-based feature selection reduces the original feature space by removing redundant attributes using a correlation threshold of 0.85. The autoencoder then learns nonlinear representations from the selected features. Finally, the FT-Transformer treats the selected behavioral features as feature-level tokens and uses self-attention to model their interactions. Finally, the Nadam optimizer is leveraged to optimize model parameters.

#### 3.1. Data Acquisition and Preprocessing

Dataset collection involves gathering relevant data from reliable sources for model development and analysis. Data preprocessing involves cleaning, transforming, and preparing the collected data to improve quality and ensure effective model performance.

##### 3.1.1. Dataset Description

The proposed TBAF-GED technique has been conducted using the Player Engagement Analysis and Prediction dataset [19]. This dataset offers a comprehensive view of demographics, player behavior, and engagement in online gaming environments, thus making it valuable for player retention analysis and engagement prediction. The dataset documentation describes the data as an online gaming behavior dataset, but it does not provide evidence that it was collected from a specific commercial game or an identifiable real-world player-log system. Therefore, in this study, the dataset is considered a publicly available tabular benchmark dataset rather than longitudinal telemetry from real players. After excluding PlayerID and the target variable engagement level, the remaining candidate attributes were used for correlation-based feature selection. The CFS procedure retained ten behavioral attributes: Age, Gender, GameGenre, PlayTimeHours, InGamePurchases, GameDifficulty, SessionsPerWeek, AvgSessionDurationMinutes, PlayerLevel, and AchievementsUnlocked. The target variable, engagement level, is provided as a three-class categorical variable with Low, Medium, and High engagement levels. The available dataset

documentation does not specify a deterministic mathematical rule for generating these categories. Therefore, this study uses engagement level as the provided target label and does not redefine or reconstruct the labeling procedure.

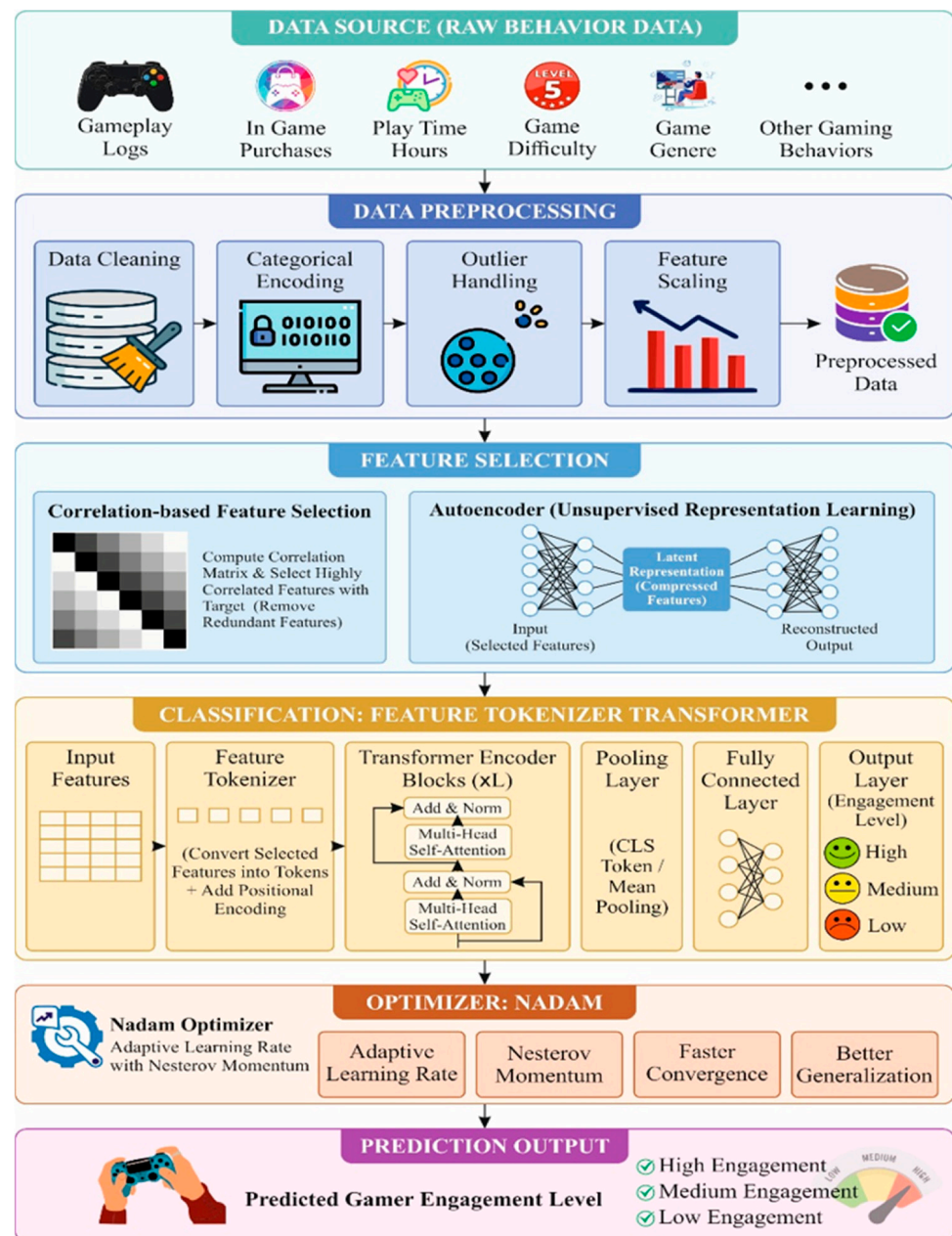


**Figure 1.** End-to-end architecture of the TBAF-GED approach.

Table 1 presents three engagement-level classes, High, Medium, and Low, with a total of 40,034 samples. These features are related to the engagement classification due to the fact that they depict various aspects of player interaction and progression. Player level and unlocked accomplishments offer details related to development and participation, while session frequency reflects the regularities of player interaction. The investigation of these features allows the proposed model to recognize relationships among behavioral characteristics that may be related to various stages of engagement. The chosen dataset directly offers behavioral and engagement-related features needed to examine the stated research question, comprising player progression, achievement, session frequency, and engagement-level information.

**Table 1.** Details of the dataset.

| Class Labels | Class Count |
|--------------|-------------|
| Low          | 10,324      |
| Medium       | 19,374      |
| High         | 10,336      |
| Total        | 40,034      |

### 3.1.2. Data Preprocessing

The data preprocessing step involves data cleaning, label encoding, outlier handling, and robust scaling to ensure the quality and reliability of the data [20].

**Data Cleaning:** Data cleaning is an important preprocessing step that involves identifying and handling missing, unreliable, or noisy values in the features. Irrelevant entries and duplicate records are removed to improve the quality of the data.

**Categorical Encoding (label encoding):** Categorical features are converted into numerical values using label encoding. Each category is assigned a different integer value so that the data can be processed by machine learning models.

**Outlier Handling:** Outliers are identified using statistical methods such as the interquartile range (IQR). Values that deviate considerably from the normal range are removed or limited to reduce their influence on the model.

**Robust Scalar:** Robust scaling is a feature standardization method that uses statistics that are less sensitive to outliers. It scales the values based on the median and IQR, making it suitable for data containing extreme values. The transformation used for robust scaling is given in Equation (1).

$$x' = \frac{x - \text{median}(x)}{IQR} \quad (1)$$

As shown in Equation (1), the median is subtracted from ( $x$ ) to center the data, and the result is divided by the IQR to reduce the effect of outliers and scale the data. This allows extreme values to have less influence on the scaling process.

### 3.2. Correlation-Based Feature-Selection Strategy

CFS-AE is applied to reduce the dimensionality of the dataset by removing redundant and irrelevant features. It also enhances model efficiency by learning compact and informative feature representations [21].

**Autoencoders:** AEs are an unsupervised learning method that encodes input values into a compressed hidden representation and decodes it to rebuild an estimation of the real input data. The autoencoder consists of an encoder and a decoder. The encoder converts the selected behavioral feature vector into a 64-dimensional latent representation, and the decoder reconstructs the original selected feature vector from this representation. The 64-dimensional latent representation is not used as an additional dimensionality-reduction step because the selected feature subset contains only ten attributes at most. Instead, it provides a nonlinear representation of the features and helps capture higher-order dependencies among the selected behavioral attributes.

The encoding operation of the autoencoder is represented by Equation (2).

$$L = f(WH + b) \quad (2)$$

In addition, the decoding is the process between the output and hidden layers. Within the context of gamer behavior reconstruction, the decoding captures the compact behavioral

representation  $L$  and rebuilds a higher-dimensional feature embedding similar to the real input data  $H$ .

The decoding operation used to reconstruct the input data is given in Equation (3).

$$X = g(W'L + b') \quad (3)$$

In Equations (2) and (3):  $f$  and  $g$  imply the activation functions,  $W$  and  $b$  signify the encoding's weights and bias,  $W'$  and  $b'$  represent the decoding's weights and bias.

**Correlation-based Feature Selection:** This technique is used in gamer user behavior classification methods to decide the most important behavioral features by employing a search approach that includes backward reduction, genetic methods, directional search, forward selection, and best-fit approaches. It is a broadly accepted model for improving the performance of the method by choosing the most informative attributes for precise engagement forecasting. The FS method is represented as follows:

The correlation-based feature selection measure is defined in Equation (4).

$$R_{xc} = \frac{nR_{xi}}{\sqrt{n + n(n-1)R_{ii}}} \quad (4)$$

Let  $R_{xc}$  denote the correlation that occurs among the total subset of attributes and class variables;  $n$  implies the subset of attributes;  $R_{xi}$  represents the average correlation that occurs among the subset of attributes and the class variables; and  $R_{ii}$  signifies the inter-correlation average that occurs among subsets of features.

Although the method might still suffer from reconstruction loss owing to noisy higher-dimensional behavioral information, redundancy between correlated attributes, overfitting because of extreme model difficulty, and minimized interpretability impacted by dense hidden representations, the following unified loss function is minimized:

The unified loss function used to address these issues is given in Equation (5).

$$\mathcal{L} = \|H - X\|_2^2 + \lambda_1 \sum_{i,j \in S} (1 - R_{ij}^c) + \lambda_2 \|W\|_F^2 + \lambda_3 \|L\|_1 \quad (5)$$

The correlation threshold of 0.85 used in CFS is a predefined feature-selection criterion and is different from the correlation-related regularization term in Equation (5). The 0.85 threshold is used to identify and remove highly redundant candidate features during feature selection, while the correlation term in Equation (5) is used during the optimization of the learned representation. Therefore, feature selection and correlation-aware representation learning have different roles in the proposed framework.

### 3.3. Gamer Engagement Classification Module

The FT-Transformer takes the selected original behavioral features obtained after correlation-based feature selection, rather than the individual dimensions of the 64-dimensional autoencoder latent representation. Each retained numerical behavioral feature is separately converted into a feature embedding using the numerical feature tokenizer. These feature embeddings are then used as input tokens to the Transformer encoder. This allows self-attention to model the interactions among the retained behavioral attributes while keeping the original meaning of each feature.

The FT-Transformer is employed as the classification model for gamer engagement detection. It classifies engagement levels by modeling complex interactions among behavioral features using self-attention [22]. The FT-Transformer is applied for classifying player engagement employing behavioral features. The method is intended to manage tabular behavioral data and identify engagement via a numerical feature tokenizer, an

MLP classifier head, and a self-attention-driven feature relationship model. A Transformer encoding method encodes global interactions between attributes. The [CLS] outcome is fed to an MLP to forecast engagement level, such as lower, medium, or higher engagement. In this context, the key concept of FT-Transformer is to consider every numerical attribute as an independent token. With the input instances  $x = [x_1, x_2, \dots, x_d] \in \mathbb{R}^d$ , all numerical attributes  $x_i$  are mapped into feature tokenization:

The feature tokenization process is defined in Equation (6).

$$t_i = x_i W_i + b_i + e_i, i = 1, 2, \dots, d \quad (6)$$

In this Equation (6),  $W_i \in \mathbb{R}^k$  and  $b_i \in \mathbb{R}^k$  signify the trainable projection embedding and bias term,  $e_i \in \mathbb{R}^k$  implies the trainable vector features, and  $k$  indicates the token dimension. For each feature token, a trainable classifier token  $t_{cls}$  is proposed to build the resulting input sequence of the token.

The input token sequence formed using the classifier token is given in Equation (7).

$$T_0 = [t_{cls}; t_1; t_2; \dots; t_d] \quad (7)$$

Here,  $t_{cls}$  represents the trainable classifier token,  $T_0 \in \mathbb{R}^{((d+1) \times k)}$  signifies the matrix of input tokens, and “;” implies concatenation with sequence size. Let  $d$  indicate the count of behavioral data, and  $k$  denotes the size of the token. The constructed sequence is then passed to a multi-layer Transformer encoder to train global dependencies between behavioral patterns.

The scaled dot-product attention operation used by the Transformer is defined in Equation (8).

$$Attn(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (8)$$

The multi-head attention operation is expressed in Equation (9).

$$MultiHead(Q, K, V) = Concat(head_1, head_2, \dots, head_h)W^O \quad (9)$$

Here,  $Q$  implies the query matrix,  $K$  indicates the key matrix, and  $V$  signifies the value matrix; correspondingly,  $d_k$  represents the key vector size,  $head_i$  implies the outcome of the  $i$ -th attention head,  $W^O$  refers to the matrix of outcome projection, and  $h$  implies the count of attention heads. The output of each Transformer encoding layer is represented in Equation (10).

$$T_l = EncoderLayer(T_{l-1}), l = 1, 2, \dots, L \quad (10)$$

Here,  $T_l$  implies the outcome of the  $l$ -th layer,  $T_{l-1}$  denotes the prior layer values, and  $L$  signifies the overall count of encoding layers. Finally, the performance of the CLS token is utilized as the global behavioral features. The engagement class probabilities obtained from the MLP classifier are calculated using Equation (11).

$$p = softmax(MLP(h_{cls})) \quad (11)$$

Now,  $MLP(\cdot)$  represents the multilayer perceptron classifier head,  $h_{cls}$  denotes the resulting latent representation of the CLS token, and  $p$  implies the probability function over engagement labels. The final engagement class is determined using the maximum predicted probability, as given in Equation (12).

$$\hat{y} = argmax(p) \quad (12)$$

Therefore, the method achieves engagement identification for all gamer behavior instance that depends on the global representation combined by the CLS token. The challenge of FT-Transformer is its reduced robustness to class imbalance in engagement conditions that might bias detection to the majority categories. The weighted cross-entropy loss used to address class imbalance is defined in Equation (13).

$$\mathcal{L} = - \sum_{c=1}^C w_c y_c \log(p_c) \quad (13)$$

In this Equation (13),  $C$  represents the count of engagement classes,  $w_c$  implies the class weight,  $p_c$  refers to the forecasted probability, and  $y_c$  implies the actual label. Based on the class distribution in the training set, the class weights for Low, Medium, and High engagement were approximately 1.293, 0.689, and 1.291, respectively. The Low and High classes were assigned higher weights because they had fewer samples than the Medium class. This weighting module allocates high significance to minority engagement categories, ensuring balanced training across classes. So, it enhances detection consistency and decreases bias to dominant engagement states in training. Figure 2 shows the architecture of the FT-Transformer-based classification model.

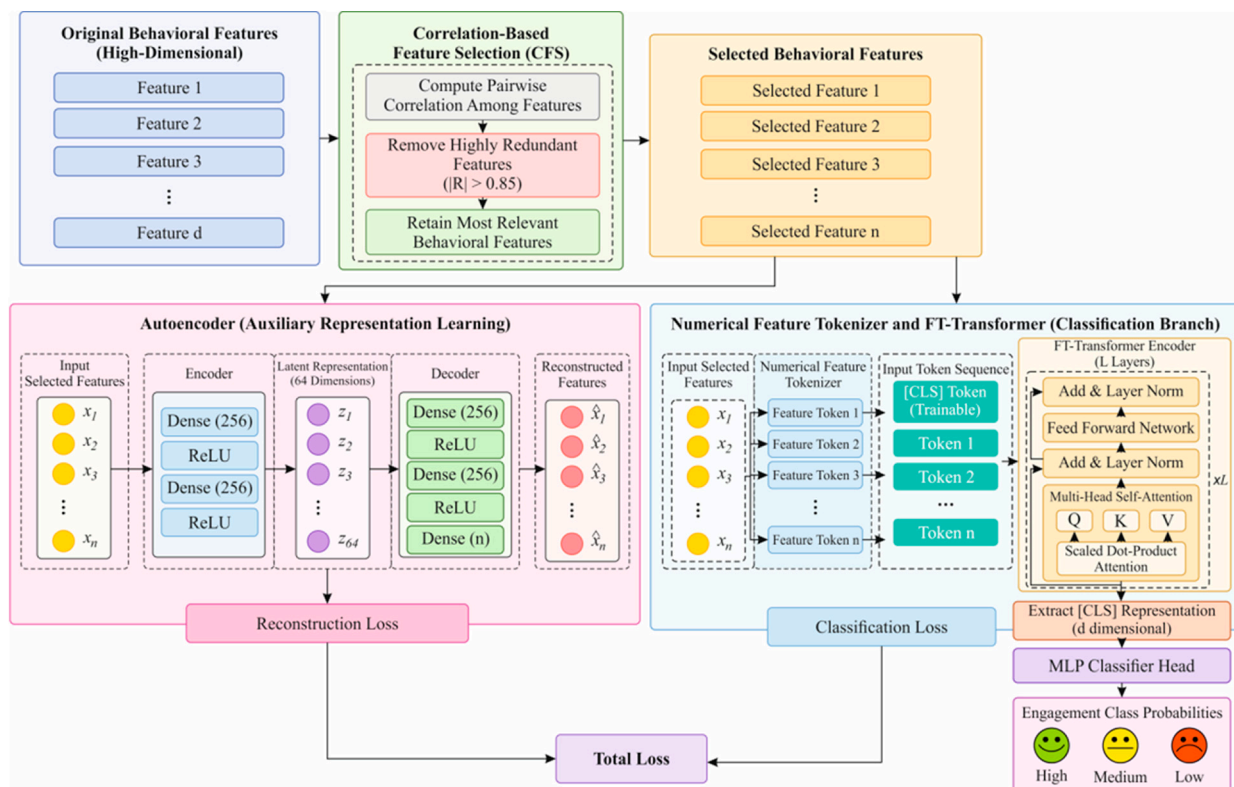


**Figure 2.** FT-Transformer-based classification framework.

### 3.4. Parameter Optimization

The Nadam optimizer is an advanced optimization algorithm that combines Adam with Nesterov momentum. It is used to accelerate convergence and improve the stability and efficiency of model training [23]. The Nadam optimizer algorithm is used to improve the training effectiveness and prediction performance of engagement classifier tasks. This optimizer integrates the adaptive learning ability of the Adam with Nesterov momentum (NAG) to enable stable convergence when learning difficult gaming patterns from behav-

ioral data. The temporary parameter update used in the Nadam look-ahead step is given in Equation (14).

$$\theta_{temp} = \theta_{t-1} - \beta_1 * \frac{m_{t-1}}{1 - \beta_1^{t-1}} \quad (14)$$

Afterward, the gradient has been measured at the point instead of the temporary value to prevent an increase in redundant computation.

Nadam performs look-ahead through mathematical evaluation. The Nesterov momentum approximation used by Nadam is defined in Equation (15).

$$\hat{m}_t^{nesterov} = \beta_1 \cdot \frac{m_t}{1 - \beta_1^t} + (1 - \beta_1) \cdot \frac{g_t}{1 - \beta_1^t} \quad (15)$$

This formula estimates the momentum condition at the look-ahead value  $\theta_{temp}$  for weighting the recent bias-corrected momentum  $\hat{m}_t^{nesterov}$  with  $\beta_1^{t+1}$ , which implies the decayed coefficient of another stage, then  $g_t$  denotes the recent gradient, and this approximation of the look-ahead momentum substitutes the Adam momentum estimation. The final parameter update performed by Nadam is given in Equation (16).

$$\theta_t = \theta_{t-1} - \alpha \cdot \frac{\hat{m}_t^{nesterov}}{\sqrt{V_t} + e'} \quad (16)$$

In this Equation (16),  $v_t$  signifies the 2nd-moment estimation, and its bias correction follows the same expression as Adam. Nadam incorporates adaptive learning rates with NAG, making it efficient for heterogeneous and higher-dimensional gaming behavior information.

Table 2 portrays the hyperparameter settings employed across various phases. The model utilizes CFS with a correlation threshold of 0.85 to eliminate extremely correlated features and decrease redundancy. An autoencoder with dropout of 0.2, a latent dimension of 64, and 2 hidden layers (256 neurons each) has been employed for nonlinear feature compression. A Feature Tokenizer Transformer with dropout 0.2, embedding dimension 128, feed-forward dimension of 512 with four Transformer layers, and 4 attention heads is utilized for feature representation learning. Training is accomplished by implementing the Nadam optimizer with a learning rate of 0.001 ( $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ). The model is trained with a batch size of 32 for 25 epochs, and early stopping with a patience of 10 is employed to avoid overfitting.

**Table 2.** Parameter settings.

| Method                                    | Parameters                   | Values |
|-------------------------------------------|------------------------------|--------|
| Correlation-Based Feature Selection (CFS) | Correlation Threshold        | 0.85   |
|                                           | Hidden Layers                | 2      |
| Autoencoder                               | Latent Dimension             | 64     |
|                                           | Hidden Neurons               | 256    |
|                                           | Dropout Rate                 | 0.2    |
| Feature Tokenizer Transformer             | Number of Transformer Layers | 4      |
|                                           | Embedding Dimension          | 128    |
|                                           | Attention Heads              | 4      |
|                                           | Feed Forward Dimension       | 512    |
|                                           | Dropout Rate                 | 0.2    |

**Table 2.** *Cont.*

| Method                 | Parameters              | Values |
|------------------------|-------------------------|--------|
| Nadam Optimizer        | Learning Rate           | 0.001  |
|                        | Beta <sub>1</sub>       | 0.9    |
|                        | Beta <sub>2</sub>       | 0.999  |
| Training Configuration | Batch Size              | 32     |
|                        | Epochs                  | 25     |
|                        | Early Stopping Patience | 10     |

The hyperparameter values involved in the proposed model are provided in Table 2. The dataset was split into training and testing subsets, and the test set was kept completely separate from the hyperparameter-selection process. The selected configuration was then used for the final training and evaluation of the proposed TBAF-GED model.

### 3.5. Model Evaluation Metrics

In this research, we used several performance metrics to assess our presented method, such as accuracy, recall, F-score, precision, and Matthews correlation coefficient (MCC), which were all discussed in the examination. A confusion matrix is a table that defines how well a categorization approach performs, and it summarizes and visualizes the prediction outcomes for a classification issue. In a confusion matrix, true negative (*TN*) signifies a value that is correctly predicted as negative, true positive (*TP*) denotes a value that is correctly predicted as positive, false negative (*FN*) illustrates a value incorrectly predicted as negative, and false positive (*FP*) represents a value incorrectly predicted as positive. Mathematically, all these evaluation metrics can be represented as:

$$Accuracy = \left( \frac{TP + TN}{TP + TN + FP + FN} \right) \times 100\% \quad (17)$$

$$Recall = \frac{TP}{TP + FN} \quad (18)$$

$$Precision = \frac{TP}{TP + FP} \quad (19)$$

$$F\ score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \quad (20)$$

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \quad (21)$$

## 4. Findings and Discussions

The performance analysis of the proposed TBAF-GED model is inspected using the Player Engagement Analysis and Prediction dataset. Table 3 illustrates the hardware and software configuration of the TBAF-GED methodology. In hardware, it utilizes an Intel Core i7 (13th Gen) along with an NVIDIA GeForce RTX 5060 GPU with 8GB VRAM and 1 TB storage space to handle the dataset, files and models. The software provides a Windows 11 (64-bit) OS along with the Python 3.11 (Python Software Foundation, Wilmington, DE, USA) programming language and PyTorch 2.11.0 for model training. To enable GPU acceleration, CUDA 12.8 is used. For data processing, NumPy 1.26.4, Pandas 2.2.3, and Scikit-learn 1.5.2 were used to perform machine learning operations. Meanwhile, Matplotlib 3.9.2 and Seaborn 0.13.2 were employed for data visualization, whereas Visual Studio Code 1.95.3 was used as the primary development environment.

**Table 3.** Hardware and software configuration of the TBAF-GED system.

| Category | Parameter               | Suggested Configuration     |
|----------|-------------------------|-----------------------------|
| Hardware | Processor (CPU)         | Intel Core i7 (13th Gen)    |
|          | GPU                     | NVIDIA GeForce RTX 5060     |
|          | GPU Memory              | 8 GB VRAM                   |
|          | RAM                     | 32 GB                       |
|          | Storage                 | 1 TB NVMe SSD               |
| Software | Operating System        | Windows 11 (64-bit)         |
|          | Programming Language    | Python 3.11                 |
|          | Deep Learning Framework | PyTorch 2.11.0              |
|          | CUDA                    | CUDA 12.8                   |
|          | Data Processing         | NumPy, Pandas, Scikit-learn |
|          | Visualization           | Matplotlib, Seaborn         |
|          | Development Environment | Visual Studio Code          |

The correlation heatmap shown in Figure 3 illustrates the features used for gamer engagement-level classification. The heatmap demonstrates the relation amongst player activity, progression, and achievement-related attributes. The correlation analysis revealed several strong positive correlations among the candidate behavioral features. Features exhibiting redundancy above the predefined 0.85 threshold were subsequently considered during correlation-based feature selection. Therefore, the strong correlations observed in the initial correlation analysis characterize the candidate feature space and should not be interpreted as indicating that all retained features have pairwise correlations above 0.85.

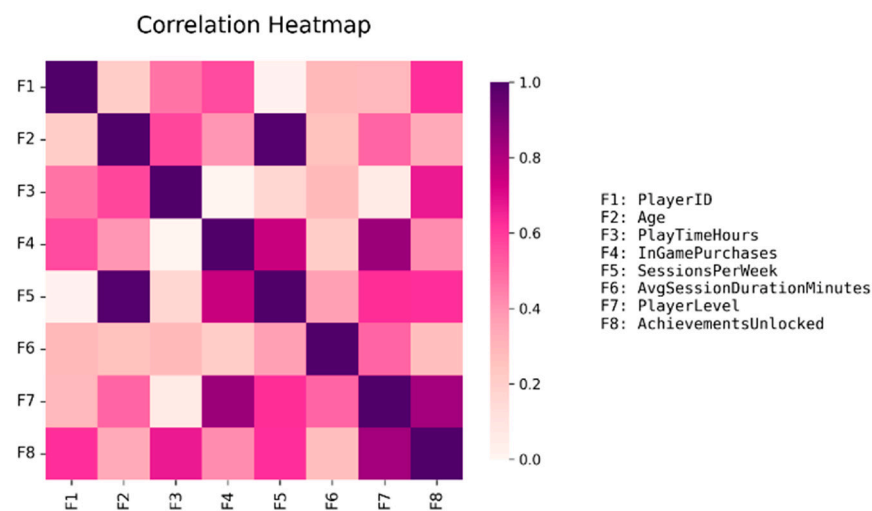
**Figure 3.** Correlation heatmap for player engagement-level classification.

Figure 4 illustrates the exploratory data analysis of video game player engagement through multiple visualizations. The dataset includes players from various age groups of both genders, which supports balanced analysis. The scatter plot highlights the interaction amongst players' level and achievements unlocked, which reflects players exhibiting high participation. Furthermore, the analysis of weekly sessions reveals that highly engaged players maintain regular participation in gaming activities compared to low- and medium-level engagement players.

## Exploratory Data Analysis of Video Game Player Engagement Dataset

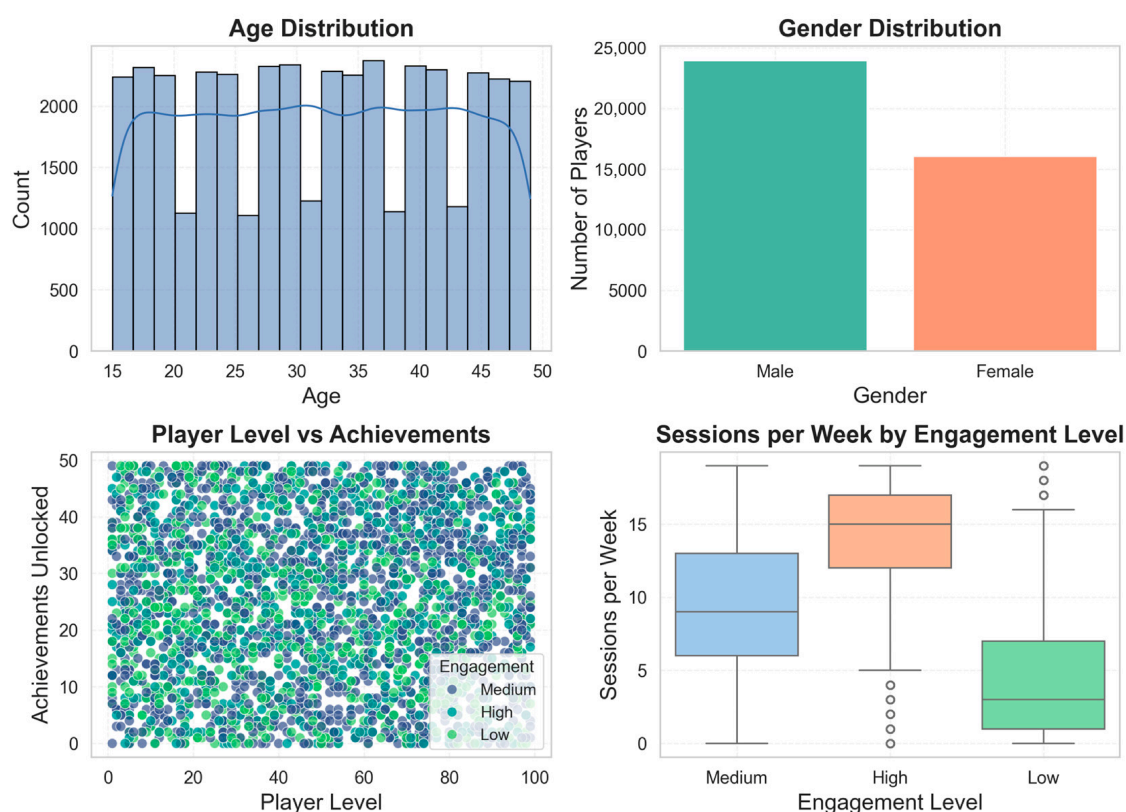


**Figure 4.** Exploratory data analysis of the proposed model.

Figure 5 shows the classifier result of the proposed model using 80% of the training data. Figure 5a displays the confusion matrix; the majority of classes are predicted correctly with minimal classification error, which indicates the effectiveness of the proposed model. Figure 5b represents the PR curve analysis, which exhibits strong predictive performance.

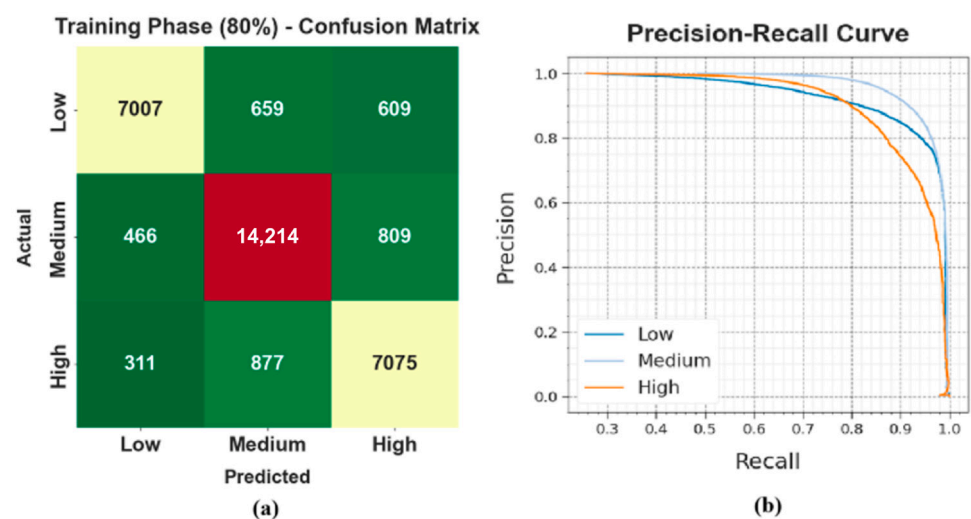


**Figure 5.** Classifier results of the TBAF-GED framework with (a) confusion matrix and (b) PR curves for 80% training.

Table 4 summarizes the player engagement-level results with 80% of the training set. The training phase indicates strong classification performance across all metrics and

achieved an average accuracy of 92.23% with balanced precision, recall, and F1-Score values, which indicates consistent class discrimination. The average MCC value of 81.49% further validates the stable and reliable performance of the proposed model during the training phase, while the overall accuracy of 88.35% indicates the model's performance across all three engagement levels.

**Table 4.** Engagement-level classifier results of the TBAF-GED model with 80% training.

| Training Phase (80%) |          |           |        |                |           |
|----------------------|----------|-----------|--------|----------------|-----------|
| Class Labels         | Accuracy | Precision | Recall | Macro F1-Score | Macro MCC |
| Low                  | 93.61    | 90.02     | 84.68  | 87.27          | 83.08     |
| Medium               | 91.22    | 90.25     | 91.77  | 91             | 82.45     |
| High                 | 91.86    | 83.3      | 85.62  | 84.45          | 78.95     |
| Average              | 92.23    | 87.86     | 87.36  | 87.57          | 81.49     |
| Overall Accuracy     | 88.35    |           |        |                |           |

Figure 6 presents the classifier results of the TBAF-GED method under the confusion matrix and ROC curve for 20% of the test set. Figure 6a reveals the confusion matrix of the TBAF-GED method that correctly identified engagement level with 1707 samples as lower, 3554 samples as medium and 1778 samples as higher. Likewise, Figure 6b shows that the ROC inspection of the TBAF-GED system offers optimal classification for each class.

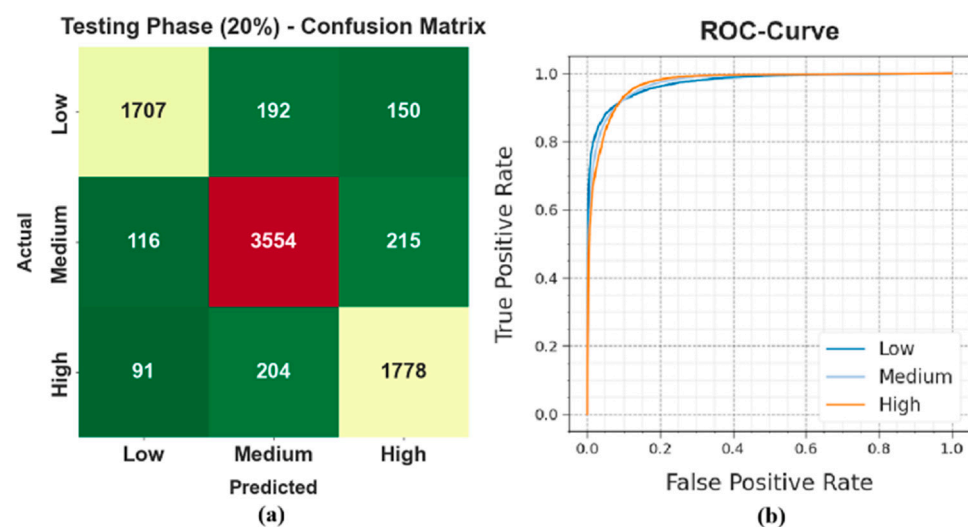


**Figure 6.** Classifier result of the TBAF-GED method with (a) confusion matrix and (b) ROC curves for the 20% testing.

Table 5 displays the player engagement-level classifier outcome of the proposed model using 20% testing. It is evident that the proposed model reaches excellent performance by achieving an average accuracy of 91.94% and an average F1-Score value of 87.07%, which highlights the effective classification capability of the model. The close alignment between precision and recall further validates the balanced classification across all classes. Moreover, the average MCC value of 80.77% confirms the stability and robustness of the proposed model during the testing phase. The model achieves an overall accuracy of 87.91% across the three engagement levels.

**Table 5.** Engagement-level classifier results of the TBAF-GED model with 20% testing.

| Class Labels     | Testing Phase (20%) |           |        |                |           |
|------------------|---------------------|-----------|--------|----------------|-----------|
|                  | Accuracy            | Precision | Recall | Macro F1-Score | Macro MCC |
| Low              | 93.14               | 89.18     | 83.31  | 86.15          | 81.68     |
| Medium           | 90.92               | 89.97     | 91.48  | 90.72          | 81.84     |
| High             | 91.76               | 82.97     | 85.77  | 84.35          | 78.77     |
| Average          | 91.94               | 87.38     | 86.85  | 87.07          | 80.77     |
| Overall Accuracy | 87.91               |           |        |                |           |

Figure 7 displays the classifier outcome of the TBAF-GED method with a confusion matrix and PR curve of the 70% training. Figure 7a shows the confusion matrix of the TBAF-GED technique, which precisely categorized engagement level with 6530 instances as low, 12,847 instances as medium and 6335 instances as high. Also, Figure 7b portrays the PR analysis of the TBAF-GED method, thus representing maximal performance at all classes.

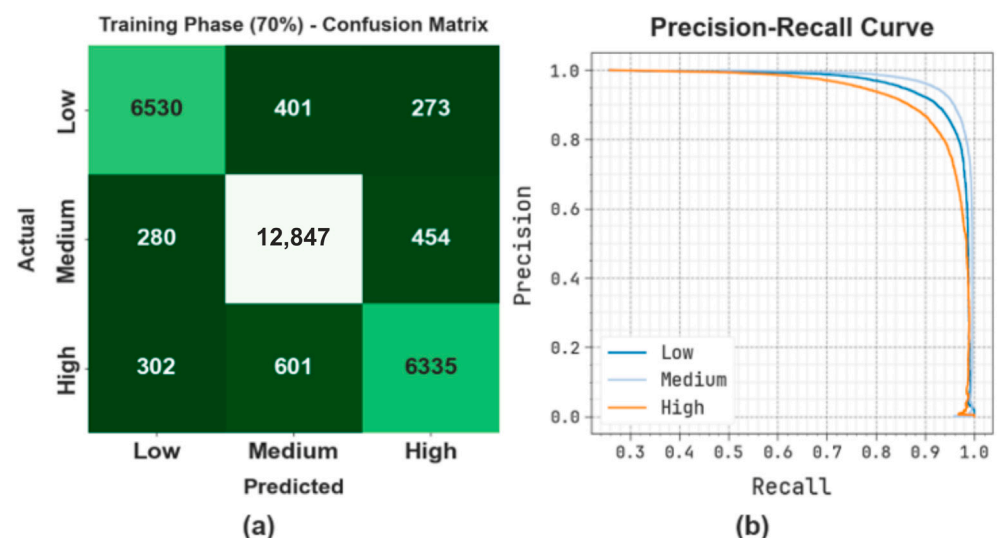
**Figure 7.** Classifier results of the TBAF-GED method with (a) confusion matrix and (b) PR curves for the 70% of training.

Table 6 represents the engagement-level classifier outcome of the TBAF-GED approach with 70% of the training data. The proposed TBAF-GED technique achieved an average accuracy of 94.50% and MCC of 86.85%, representing consistent and strong classification performance. The overall accuracy obtained across all three engagement levels is 91.75%.

**Table 6.** Engagement-level classifier results of the TBAF-GED approach with 70% training.

| Class Labels         | Accuracy | Precision | Recall | Macro F1-Score | Macro MCC |
|----------------------|----------|-----------|--------|----------------|-----------|
| Training Phase (70%) |          |           |        |                |           |
| Low                  | 95.52    | 91.82     | 90.64  | 91.23          | 88.22     |
| Medium               | 93.81    | 92.76     | 94.60  | 93.67          | 87.62     |
| High                 | 94.18    | 89.71     | 87.52  | 88.60          | 84.71     |
| Average              | 94.50    | 91.43     | 90.92  | 91.17          | 86.85     |
| Overall Accuracy     | 91.75    |           |        |                |           |

Figure 8 shows the classifier outcome of the TBAF-GED approach with the confusion matrix and ROC curve of the 30% training set. Figure 8a presents the confusion matrix of the TBAF-GED model, which accurately classified engagement level with 2826 instances as low, 5463 instances as medium and 2687 instances as high. Also, Figure 8b shows the ROC analysis of the TBAF-GED method, demonstrating that the system provides effective classification across all three classes.

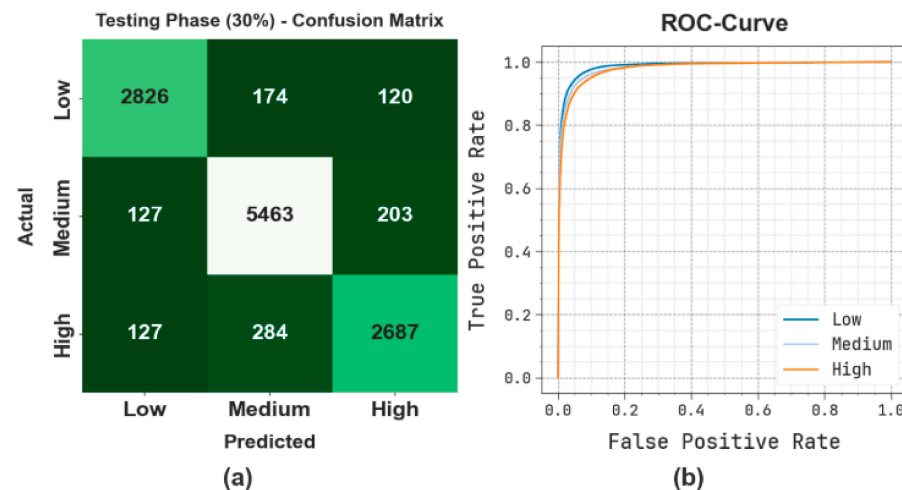


**Figure 8.** Classifier result of the TBAF-GED method with (a) confusion matrix and (b) ROC curves for the 30% testing.

Table 7 shows the engagement-level classifier outcome of the TBAF-GED system with 30% testing. The proposed TBAF-GED technique achieved an average accuracy of 94.26% and MCC of 86.29%, presenting consistent and high-quality classification outcomes. The model attains an overall accuracy of 91.38% for the three engagement levels.

**Table 7.** Engagement-level classifier results of the TBAF-GED approach with 30% testing.

| Class Labels        | Accuracy | Precision | Recall | Macro F1-Score | Macro MCC |
|---------------------|----------|-----------|--------|----------------|-----------|
| Testing Phase (30%) |          |           |        |                |           |
| Low                 | 95.44    | 91.75     | 90.58  | 91.16          | 88.09     |
| Medium              | 93.44    | 92.26     | 94.30  | 93.27          | 86.89     |
| High                | 93.89    | 89.27     | 86.73  | 87.98          | 83.90     |
| Average             | 94.26    | 91.10     | 90.54  | 90.81          | 86.29     |
| Overall Accuracy    | 91.38    |           |        |                |           |

Figure 9 illustrates the classifier performance of the TBAF-GED model by offering the accuracy and loss analyses. The accuracy curves show that the TBAF-GED method achieves gradually higher values as the number of epochs rises, considering efficient learning and convergence. Likewise, the validation accuracy closely followed the training accuracy, signifying that the model generalizes better to unseen data. The loss curves show that the training and validation losses converge to significantly correlated values. This closer alignment between training and validation loss further ensures that the TBAF-GED system learns efficiently and demonstrates stronger generalization ability with lower overfitting.

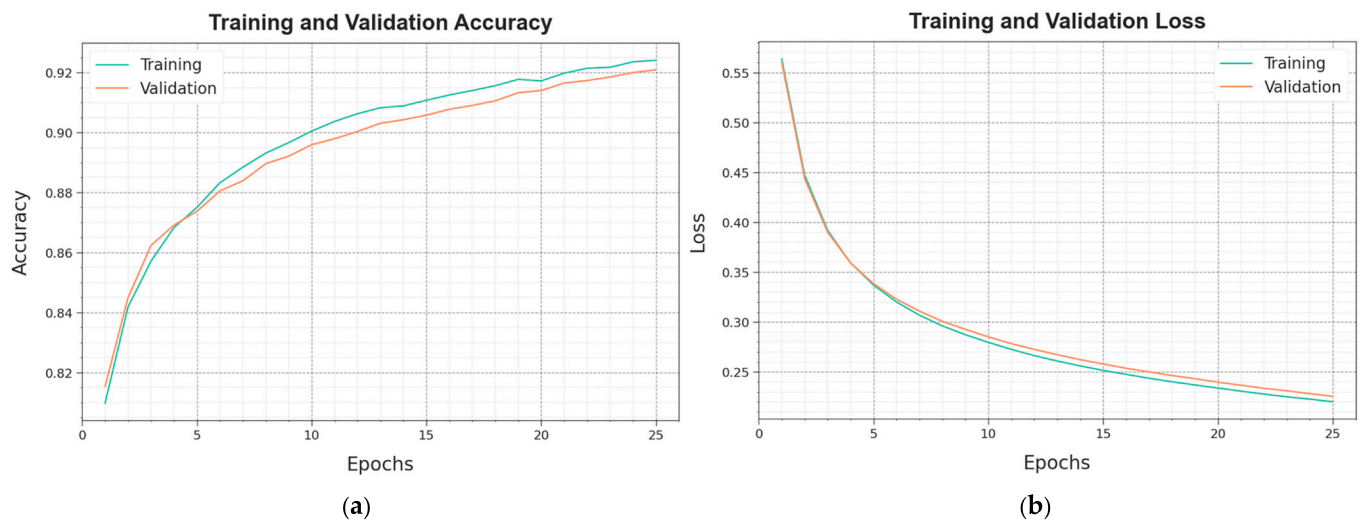


**Figure 9.** Classifier results of the TBAF-GED method with (a) accuracy and (b) loss curves.

Table 8 depicts the comparison outcome of the TBAF-GED approach with other methodologies using evaluation metrics [12,24–27]. To ensure fair comparison, the baseline models were re-implemented and validated on the same dataset using the same preprocessing and evaluation protocol, rather than directly adopting performance values in the literature. The results demonstrate that the ResNet18+NN, LSTM and Decision Tree methodologies attain ineffective performance, whereas the Random Forest, LightGBM, XGBoost, TabTransformer and 1D-CNN models gain somewhat greater performance. Moreover, the TabNet and FT-Transformer approaches achieved considerable outcomes. Finally, the proposed TBAF-GED framework achieves promising performance with an accuracy of 91.38%, precision of 91.10%, recall of 90.54% and F1-score of 90.81%, respectively. The results demonstrate the consistent classification performance of the proposed system compared with the other evaluated methodologies.

**Table 8.** Comparative results of the TBAF-GED method with existing approaches.

| Methods        | Accuracy | Precision | Recall | F1-Score |
|----------------|----------|-----------|--------|----------|
| FT-Transformer | 91.16    | 91.02     | 90.47  | 90.74    |
| 1D-CNN         | 90.84    | 89.96     | 89.42  | 89.69    |
| TabNet         | 91.02    | 90.36     | 89.82  | 90.09    |
| TabTransformer | 90.31    | 90.71     | 90.18  | 90.44    |
| Decision Tree  | 79.58    | 86.00     | 80.00  | 81.00    |
| Random Forest  | 89.21    | 89.00     | 89.00  | 89.00    |
| LSTM           | 78.48    | 79.73     | 78.61  | 77.83    |
| ResNet18+NN    | 72.00    | 74.32     | 71.73  | 73.11    |
| XGBoost        | 90.11    | 89.72     | 89.31  | 89.51    |
| LightGBM       | 89.72    | 89.43     | 88.96  | 89.19    |
| TBAF-GED       | 91.38    | 91.10     | 90.54  | 90.81    |

Table 9 presents the ablation study of the proposed TBAF-GED model using different evaluation measures. The model without feature selection achieves an accuracy of 88.64%, while the integration of correlation-based feature selection with FTT increases the accuracy to 89.72%. Integrating the autoencoder further improves the accuracy to 90.46%. The

combination of correlation-based feature selection, autoencoder, and FTT without Nadam achieves an accuracy of 90.92%. Finally, the TBAF-GED (Preprocessing + Correlation FS + Autoencoder + FTT + Nadam) model, incorporating preprocessing, correlation-based feature selection, autoencoder, FTT, and Nadam, achieves an accuracy of 91.38%, precision of 91.10%, recall of 90.54%, F1-score of 90.81%, and MCC of 86.29%, respectively. The results indicate that the successive integration of the model components contributes to improved classification performance.

**Table 9.** Ablation study of the TBAF-GED approach.

| Methods                                                               | Accuracy | Precision | Recall | F1-Score | MCC   |
|-----------------------------------------------------------------------|----------|-----------|--------|----------|-------|
| Without Feature Selection                                             | 88.64    | 88.21     | 87.46  | 87.82    | 78.42 |
| Correlation-Based FS Only + FTT                                       | 89.72    | 89.16     | 88.38  | 88.75    | 80.16 |
| Autoencoder Only + FTT                                                | 90.46    | 89.84     | 89.12  | 89.46    | 82.04 |
| Correlation-Based FS + Autoencoder + FTT (without Nadam)              | 90.92    | 90.48     | 89.86  | 90.14    | 84.12 |
| TBAF-GED (Preprocessing + Correlation FS + Autoencoder + FTT + Nadam) | 91.38    | 91.10     | 90.54  | 90.81    | 86.29 |

Table 10 presents a comparison of the computational complexity of the TBAF-GED model and other deep learning models based on parameters, FLOPs, and inference time. The TBAF-GED model requires the fewest parameters at 0.31 M, with the lowest computational cost of 5.20 M FLOPs and an inference time of 0.42 ms. In comparison, 1D-CNN, TabNet, LSTM, and TabTransformer record higher computational complexity than TBAF-GED. These results indicate that TBAF-GED maintains a lower computational complexity among the evaluated models.

**Table 10.** Computational complexity performance of the proposed TBAF-GED model.

| DL Model       | Parameters (M) | FLOPs (M) | Inference Time (ms) |
|----------------|----------------|-----------|---------------------|
| 1D-CNN         | 0.68           | 18.60     | 0.96                |
| TabNet         | 1.24           | 32.80     | 1.48                |
| LSTM           | 0.91           | 24.70     | 1.21                |
| TabTransformer | 2.16           | 46.30     | 1.87                |
| TBAF-GED       | 0.31           | 5.20      | 0.42                |

Table 11 demonstrates the K-fold validation analysis of the TBAF-GED approach. The k-7 fold recorded the highest accuracy of 91.24%, while the lowest result was observed in k-1 with an accuracy of 89.87%. The accuracy values ranged from 89.87% to 91.24%, signifying consistent classification outcomes across the distinct folds. The average values were an accuracy of 90.54%, precision of 90.00%, recall of 89.57% and F1-Score of 89.79%. The corresponding SD values were 0.40, 0.39, 0.39, and 0.37 respectively. Hence, the low SD values suggest that the TBAF-GED system remains stable in terms of outcomes under ten folds.

Table 12 represents the Wilcoxon test performance of the TBAF-GED system using the accuracy metric. Across every four comparisons, the Wilcoxon statistic was  $W = 0$ , along with 0.00195  $p$ -value. However, because the  $p$ -value is below the 0.05 significance level,

the observed variance is statistically significant. This reliable pattern was observed for comparisons of TBAF-GED vs. without feature selection with correlation- and autoencoder-based methodologies. Finally, the TBAF-GED vs. Correlation-Based FS + Autoencoder + FTT (without Nadam) framework records stable and consistent performance with significant accuracy.

**Table 11.** K-Fold cross-validation analysis of the proposed TBAF-GED method.

| K-Fold | Accuracy | Precision | Recall | F1-Score |
|--------|----------|-----------|--------|----------|
| k-1    | 89.87    | 89.34     | 88.92  | 89.13    |
| k-2    | 90.12    | 89.68     | 89.21  | 89.44    |
| k-3    | 90.46    | 89.91     | 89.57  | 89.74    |
| k-4    | 90.28    | 89.76     | 89.38  | 89.57    |
| k-5    | 90.73    | 90.12     | 89.64  | 89.88    |
| k-6    | 90.91    | 90.38     | 89.82  | 90.10    |
| k-7    | 91.24    | 90.76     | 90.17  | 90.46    |
| k-8    | 90.64    | 90.05     | 89.71  | 89.88    |
| k-9    | 90.38    | 89.82     | 89.43  | 89.62    |
| k-10   | 90.79    | 90.21     | 89.86  | 90.03    |
| Mean   | 90.54    | 90.00     | 89.57  | 89.79    |
| SD     | 0.40     | 0.39      | 0.39   | 0.37     |

**Table 12.** Wilcoxon test results of the TBAF-GED algorithm based on the accuracy metric.

| Comparison                                                            | Metric   | Wilcoxon Statistic (W) | p-Value | Significance |
|-----------------------------------------------------------------------|----------|------------------------|---------|--------------|
| TBAF-GED vs. Without Feature Selection                                | Accuracy | 0                      | 0.00195 | Significant  |
| TBAF-GED vs. Correlation-Based FS Only + FTT                          | Accuracy | 0                      | 0.00195 | Significant  |
| TBAF-GED vs. Autoencoder Only + FTT                                   | Accuracy | 0                      | 0.00195 | Significant  |
| TBAF-GED vs. Correlation-Based FS + Autoencoder + FTT (without Nadam) | Accuracy | 0                      | 0.00195 | Significant  |

The proposed technique for gamer player engagement identification combines a stronger pipeline comprising data preprocessing, CFS-AE-driven feature extraction, FT-Transformer for classification, and Nadam optimization for effective learning. It efficiently handles noisy, higher-dimensional behavioral data by integrating association-based filtering with higher autoencoder extraction learning. The FT-Transformer improves the outcome by extracting risk feature interactions and sequential dependency in users' behavior. In comparison with standard ML techniques, the proposed architecture delivers enhanced precision and superior generalization on datasets, namely the Kaggle user engagement dataset.

## 5. Conclusion and Future Work

This research has introduced a TBAF-GED using behavioral data. The proposed model employs a CFS-AE to reduce redundancy and irrelevant features in the feature space. For gamer engagement detection, the FT-Transformer provides a novel tokenization model that

transforms behavioral features into embeddings and captures their complex interactions through self-attention. Additionally, the Nadam optimizer is utilized to update model parameters and improve convergence stability during training. Experimental evaluation demonstrates that the proposed TBAF-GED framework attains promising performance, with an accuracy of 94.26% compared with existing approaches. The proposed framework classifies players into Low, Medium, and High engagement levels based on the available gaming-related behavioral attributes. The results demonstrate the potential of feature-level Transformer modeling for engagement classification within the investigated dataset. The framework provides a computational approach for classifying engagement levels from aggregated gaming-related behavioral attributes and establishes a basis for future validation using real-world longitudinal player telemetry. It helps improve game design decisions and enhances player retention through intelligent engagement analysis. However, the proposed technique may face difficulties in scalability on extremely large datasets because of higher computational needs, and may demonstrate limited results in hugely different or unobserved gaming settings without appropriate field adaptation. In the future, it can concentrate on building lighter Transformer techniques to decrease computation and assist real-world involvement recognition. XAI approaches can be further added to create predictions that are most understandable for developers. Superior strategies, such as contrastive learning or data reinforcement, can enhance outcomes on unnecessary information. The system can be extended for cross-gaming and real-world adaptive engagement prediction to enhance practical usage.

Because the dataset provides aggregated behavioral attributes and the engagement-level target is supplied by the dataset source, potential dependence between engagement-related input variables and the target cannot be completely excluded. In particular, variables such as play frequency and session duration may be strongly associated with engagement classification. Therefore, the reported predictive performance should be interpreted as classification performance on the supplied benchmark dataset rather than as independent evidence of causal or real-world engagement inference. The present experiments validate the predictive ability of the TBAF-GED from the existing structured interaction features instead of directly assessing the impact of VR hardware or graphical characteristics on user experience.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data that support the findings of this study are openly available in the Kaggle repository at <https://www.kaggle.com/code/sulaniishara/player-engagement-analysis-prediction/input>, reference number [19] (Accessed on 1 August 2026).

**Conflicts of Interest:** The author declare no conflicts of interest.

## References

1. Ng, Y.; Nathan, R.J.; Khong, C.W. Affective game design elements for deep player engagement in indie games. *Univers. Access Inf. Soc.* **2026**, *25*, 15. [\[CrossRef\]](#)
2. Ciesielski, A. Behavioral Determinants of Player Engagement in Online Games: An Exploratory Data Analysis and Interpretable Machine Learning Study. 2026. Available online: [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=6023734](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=6023734) (accessed on 12 June 2026). [\[CrossRef\]](#)
3. Lei, S.; Luo, J.; Nan, G.; Li, D. From virtual to physical: The spillover effects of digital games on tourist destinations. *Tour. Manag.* **2026**, *114*, 105370. [\[CrossRef\]](#)
4. Gong, Y.; Wang, M.; He, L.; Xu, C.; Yu, Y. Asking, playing, learning: Investigating large language model-based scaffolding in digital game-based learning for elementary artificial intelligence education. *J. Educ. Comput. Res.* **2026**, *64*, 311–343. [\[CrossRef\]](#)

5. Liu, J.; Shadiev, R.; Ge, D.; Cao, M.; Xu, Y.; Zhou, R. Assessing digital etiquette in adolescents: A stealth approach using game-based learning to reduce anxiety, promote flow, and enhance learning gains. *Comput. Educ.* **2026**, *243*, 105475. [\[CrossRef\]](#)
6. Zakani, Z.; Moradi, H.; Ghasemzadeh, S.; Riazzi, M.; Mortazavi, F. Development and initial validation of FishFinder as a machine learning-Based serious video game for objective ADHD screening in children aged 5–12. *Multimed. Tools Appl.* **2026**, *85*, 150. [\[CrossRef\]](#)
7. Kravets, A.G. Depression Diagnosing Based on Analysis of Computer Game User Behavior. In *Human-Centric Cyber-Society*; Springer Nature: Cham, Switzerland, 2026; pp. 249–264. [\[CrossRef\]](#)
8. Choi, J.; Choi, Y. Unlocking Player Engagement for Game Design. In *International Conference on Human-Computer Interaction*; Springer Nature: Cham, Switzerland, 2025; pp. 278–287. [\[CrossRef\]](#)
9. Mulla, R.; Potharaju, S.; Tambe, S.N.; Joshi, S.; Kale, K.; Bandishti, P.; Patre, R. Predicting player churn in the gaming industry: A machine learning framework for enhanced retention strategies. *J. Curr. Sci. Technol.* **2025**, *15*, 103. [\[CrossRef\]](#)
10. Black, J.; Bryce, R. Player Preferences For Play of the Game: A Study of Player Engagement on Video Game Highlights. In *Advances in Computer Games*; Springer Nature: Cham, Switzerland, 2025; pp. 39–50. [\[CrossRef\]](#)
11. Rashed, A.; Shirmohammadi, S.; Hefeeda, M. Real-Time Player Engagement Measurement Using Non-Intrusive Game Telemetry. *IEEE Open J. Instrum. Meas.* **2025**, *4*, 2500116. [\[CrossRef\]](#)
12. Ozyirmidokuz, E.K.; Celik, B.A.; Stoica, E.A. # GAMEADDICTED: A Machine Learning Framework for Digital Game Addiction Detection, and Early Intervention. *IEEE Access* **2025**, *13*, 135757–135775. [\[CrossRef\]](#)
13. Yin, J.; Feng, Y.; Liu, Y. Modeling behavioral dynamics in digital content consumption: An attention-based neural point process approach with applications in video games. *Mark. Sci.* **2025**, *44*, 220–239. [\[CrossRef\]](#)
14. Wang, Y.; Lee, J.H. Game user interest prediction based on social relationships and spatio-temporal behavior. *Discov. Comput.* **2026**, *29*, 115. [\[CrossRef\]](#)
15. Liang, R.; Ye, Z.; Liang, Y.; Li, S. Deep learning-based player behavior modeling and game interaction system optimization research. In *2025 6th International Conference on Computer Engineering and Application (ICCEA)*; IEEE: Piscataway, NJ, USA, 2025; pp. 01–04. [\[CrossRef\]](#)
16. Han, Y.J.; Moon, J.; Woo, J. Prediction of churning game users based on social activity and churn graph neural networks. *IEEE Access* **2024**, *12*, 101971–101984. [\[CrossRef\]](#)
17. Lauer-Schmaltz, M.W.; Cash, P.; Hansen, J.P.; Das, N. Human digital twins in rehabilitation: A case study on exoskeleton and serious-game-based stroke rehabilitation using the ETHICA methodology. *IEEE Access* **2024**, *12*, 180968–180991. [\[CrossRef\]](#)
18. Marín-Lora, C.; Chover, M.; Martín, M.Y.; García-Rytman, L. Creating a treadmill running video game with smartwatch interaction. *Multimed. Tools Appl.* **2024**, *83*, 57709–57729. [\[CrossRef\]](#)
19. El Kharoua, R. Player Engagement Analysis & Prediction Dataset. Kaggle. 2024. Available online: <https://www.kaggle.com/code/sulaniishara/player-engagement-analysis-prediction/input> (accessed on 18 August 2026). [\[CrossRef\]](#)
20. Izonin, I.; Tkachenko, R.; Shakhovska, N.; Ilchyshyn, B.; Singh, K.K. A two-step data normalization approach for improving classification accuracy in the medical diagnosis domain. *Mathematics* **2022**, *10*, 1942. [\[CrossRef\]](#)
21. Alhassan, S.; Abdul-Salaam, G.; Micheal, A.; Missah, Y.M.; Ganaa, E.D.; Shirazu, A.S. CFS-AE: Correlation-based Feature Selection and Autoencoder for Improved Intrusion Detection System Performance. *J. Internet Serv. Inf. Secur.* **2024**, *14*, 104–120. [\[CrossRef\]](#)
22. Gorishniy, Y.; Rubachev, I.; Khrulkov, V.; Babenko, A. Revisiting deep learning models for tabular data. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 18932–18943. [\[CrossRef\]](#)
23. Liu, X.; Qi, H.; Jia, S.; Guo, Y.; Liu, Y. Recent advances in optimization methods for machine learning: A systematic review. *Mathematics* **2025**, *13*, 2210. [\[CrossRef\]](#)
24. Ntetska, A.; Mimou, A.; Tzimourta, K.D.; Angelidis, P.; Tsiouras, M.G. A Deep Learning Approach to Classifying User Performance in BCI Gaming. *Electronics* **2025**, *14*, 4974. [\[CrossRef\]](#)
25. Rashed, A.; Shirmohammadi, S.; Amer, I.; Hefeeda, M. A review of player engagement estimation in video games: Challenges and opportunities. *ACM Trans. Multimed. Comput. Commun. Appl.* **2025**, *21*, 1–33. [\[CrossRef\]](#)
26. Hirpara, A.; KumarJha, A. A Comprehensive Review on Age Prediction Using Machine Learning and Deep Learning Techniques. *Int. Res. J. Mod. Eng. Technol. Sci.* **2026**, *8*, 3644–3665.
27. Qin, F.; Hsu, T.C. Predictive Modeling of Online Game User Retention Based on Flow Theory and LightGBM. *Comput. Life* **2026**, *14*, 44–47. [\[CrossRef\]](#)

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.