Solving Problematic Internet Use with Sleep Modeling and Robust Voting

1st Le The Hien
Faculty of Information Technology
UET-VNU
Hanoi, Vietnam
22028101@vnu.edu.vn

2nd Le Van Duc
Faculty of Information Technology
UET-VNU
Hanoi, Vietnam
22028041@vnu.edu.vn

3rd Pham Hoang Hiep
Faculty of Information Technology
UET-VNU
Hanoi, Vietnam
22028005@vnu.edu.vn

***Abstract*—Child Mind Institute Kaggle competition involves predicting problematic internet use (PIU) in children using actigraphy data, which is plagued by missing values, noise, and variability. Many leaderboard solutions suffer from overfitting and data leakage. To address this, we propose a robust pipeline incorporating advanced imputation, noise reduction, and regularization to prevent overfitting. Our approach also focuses on rigorous validation, seed stability testing, and feature engineering for generalizability and fairness across leaderboards. This pipeline ensures stable, reasonable, and interpretable results, offering both high performance and valuable insights into the relationship between sleep patterns and PIU.**

***Index Terms*—Missing Data, Machine Learning, Prediction, Imputation Methods**

I. Introduction

Problematic Internet Use (PIU), affecting 20–45% globally [1], involves excessive internet use linked to depression, anxiety, ADHD, aggression, reduced physical activity, and lower life satisfaction [1]–[5]. Children and adolescents are especially vulnerable due to their developmental stage, which influences long-term health and social outcomes [6], [7].

While assessing PIU typically requires professional evaluations that many families cannot access, physical fitness metrics, such as accelerometer-measured activity levels, offer an accessible alternative. Using these indicators to predict PIU enables early interventions and evidence-based health and education policies.

The Child Mind Institute (CMI) Kaggle competition on PIU prediction attracted significant attention from the AI/ML community due to the complexity and significance of the challenge [8]. The competition dataset combines tabular data with time-series actigraphy files, capturing accelerometer readings over several days. However, the dataset is fraught with challenges, including missing labels (nearly 30% of the training data), noisy and heterogeneous data, and biases arising from systemic and random factors. These issues mirror real-world psychological research complexities, making it difficult to process and model the data effectively.

Since the competition's launch, nearly 3,500 teams have participated, publishing hundreds of notebooks. Although some high-ranking notebooks achieved impressive leaderboard scores (e.g., 0.494 and 0.497), they have been criticized for their lack of reliability, with issues such as data leakage, data drift, and overly complex or redundant code making their approaches questionable. These challenges highlight the competition's core difficulty: designing a robust pipeline to handle missing data, mitigate noise, and prevent overfitting while effectively leveraging both tabular and time-series components of the dataset.

To handle missing data, participants have used techniques like null value handling, imputation, and autoencoders. However, the large amount of missing data increases the risk of leakage, causing discrepancies between cross-validation and leaderboard scores, leading to overfitting. For time-series data, some teams use simpler methods like statistical features (mean, standard deviation, trend analysis), but these approaches may oversimplify the data's temporal structure, reducing performance when capturing complex patterns in actigraphy signals.

Therefore, we propose a solution grounded in stable, transparent, and reliable criteria. Our pipeline consistently achieves a stable score of 0.471 across multiple random seeds, demonstrating its robustness and reproducibility. Through a carefully designed feature engineering approach, the features used to train the model are closely aligned with the Problematic Internet Use (PIU) problem, incorporating both tabular data and time-series information. This ensures that the model captures the temporal dynamics of the actigraphy data, enhancing its ability to predict PIU effectively.

In this report, we introduce the SMRV (Solving PIU with Sleep Modeling and Robust Voting) solution, which effectively addresses the challenges posed by the Child Mind Institute competition on Problematic Internet Use (PIU) prediction. Our solution stands out due to its clarity, stability, and efficiency in dealing with the dataset's complexities:

- **Clear Approach**: Feature engineering is straightforward and directly relevant to the PIU problem, incorporating time-series data with sleep modeling.
- **Stable Method**: Minimizes randomness, ensuring consistent performance across multiple random seeds.
- **Efficient Performance**: Achieves strong leaderboard performance and robust cross-validation results, demonstrating high predictive accuracy and generalizability.

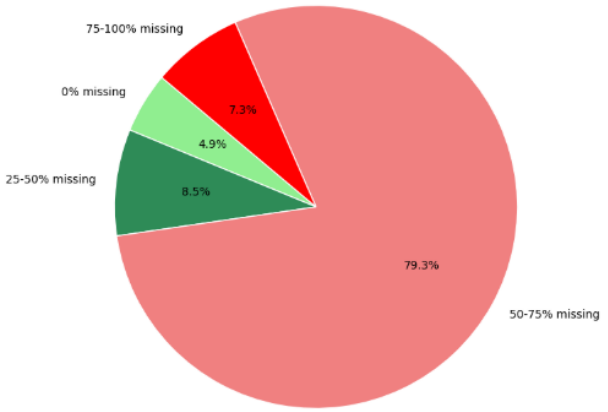

Fig. 1. The training dataset contains a significant amount of missing data, with 65 features having 50-75% missing values and 6 features suffering from 75-100% missing values, while only 4 features have complete data. This high level of missingness poses challenges for traditional imputation methods, as it can introduce bias or data leakage, necessitating more advanced approaches to handle the missing data effectively and ensure model reliability.

II. Related work

A. Feature Engineeing

1) Exploratory Data Analysis: Exploratory Data Analysis (EDA) is a critical step in data preprocessing that reveals patterns, relationships, and potential issues within a dataset. In this competition, which involves a combination of tabular and actigraphy time-series data with substantial missing values, comprehensive EDA is crucial for understanding the complexities of the data. This process informs strategies for data imputation, noise reduction, and feature extraction, helping to address challenges such as data leakage and overfitting, thus ensuring the reliability of predictive models. Previous research has demonstrated that effective EDA can guide the handling of noisy or incomplete data through techniques like imputation, outlier removal, and feature engineering, ultimately improving model performance. In competitions involving significant missing data, improper handling can lead to issues like data leakage or overfitting, which distort cross-validation results and undermine model generalization.

2) Feature Engineering Networks: Feature Engineering Networks (FEnet) are designed to enhance predictive modeling by generating new features through operations such as multiplication and division. In this approach, the network combines multiple feature generation instances, such as 'FeatGen', with different operations (e.g., 'div', 'mult'), to create new features that improve model performance. These generated features are then pooled for further processing, allowing for more effective interactions between features. FEnet focuses on feature interaction and generation, rather than relying solely on raw features, which can lead to more accurate and robust models, particularly in tasks involving complex datasets such as time-series and tabular data.

3) Detection Sleep from Time Series: Handling time-series data effectively is crucial for detecting sleep periods and generating meaningful features, particularly when using metrics such as `onset_time`, `wakeup_time`, `sleep_length`, and `sleep_enmo_mean`. In this work, we applied time-series analysis to detect sleep periods by leveraging models developed in previous competitions. These models utilize accelerometer data from wrist-worn devices, with features like `onset_time` and `wakeup_time` to mark the beginning and end of sleep episodes, respectively. The `sleep_length` feature captures the duration of the detected sleep period, while `sleep_enmo_mean` provides an average measure of activity (or lack thereof) during sleep.

By analyzing these time-series features, such as the changes in movement and light exposure (`sleep_light_mean`), we can more effectively detect periods of sleep and wakefulness. This is instrumental in identifying behavioral patterns related to Problematic Internet Use (PIU), where sleep patterns are known to have a significant influence on a person's overall health and daily routines. These features contribute to the generation of meaningful inputs for downstream tasks, offering a robust foundation for PIU prediction models.

4) Traditional Time-Series Data Processing: To effectively handle time-series data, we computed basic statistics such as the mean, max, and mode for various features in the accelerometer dataset. These features included acceleration measurements along the X, Y, and Z axes, ENMO (Euclidean Norm Minus One), anglez (arm angle relative to the horizontal plane), ambient light, and battery voltage. We also analyzed flags such as the non-wear flag, time of day, weekday, and quarter to provide temporal context. These summary statistics help in capturing key behavioral patterns and serve as compact representations of the raw data, which can be used for further analysis and feature engineering, particularly for tasks like sleep detection.

B. Models for Tabular Data

1) LightGBM : LightGBM is a gradient boosting framework that builds tree-based models, optimized for speed and memory efficiency. It handles large datasets and high-dimensional features effectively by employing histogram-based learning and leaf-wise tree growth. In this work, we utilize the LGBMRegressor with carefully tuned hyperparameters, including 300 estimators and a fixed random state for reproducibility, to achieve robust predictions on tabular data.

2) XGBoost: XGBoost (Extreme Gradient Boosting) is a widely-used machine learning model designed for scalability and flexibility. It integrates techniques like regularization and approximate tree learning to prevent overfitting while maintaining computational efficiency. We implement the XGBRegressor with custom hyperparameters tailored to our dataset, making it a reliable choice for tasks involving structured data.

3) CatBoost: CatBoost is a gradient boosting library specifically designed to handle categorical features efficiently without requiring extensive preprocessing. It employs a unique technique called ordered boosting to reduce target leakage and overfitting. Using the CatBoostRegressor, we capitalize on its

ability to natively process categorical data, ensuring seamless integration with our tabular datasets.

C. Hyperparameter Optimization

Hyperparameter optimization plays a crucial role in enhancing model performance, and various frameworks have been developed to streamline this process. Among them, Optuna has emerged as a widely-used tool due to its efficiency and flexibility. Optuna employs a dynamic sampling strategy to explore hyperparameter spaces effectively and incorporates early stopping mechanisms to prune underperforming trials, saving computation time. Its integration into machine learning pipelines has demonstrated significant improvements in model accuracy and stability across diverse domains, including time-series analysis and ensemble methods. In this work, we leverage Optuna to automate the hyperparameter tuning process, ensuring optimal performance for our models.

D. Ensemble Model

Ensemble models have gained significant attention in machine learning due to their ability to combine multiple models, thereby improving performance and generalization. These models aggregate the predictions of individual learners to produce a more robust and accurate output. In the context of sleep detection, ensemble approaches are particularly effective as they can incorporate diverse algorithms that handle different data patterns and features. One common approach is to combine models like decision trees, support vector machines, and neural networks, each contributing its strength to the overall prediction process. For instance, while decision trees can handle non-linear relationships, neural networks excel at capturing complex patterns data.

III. Proposed Method

In our approach "Fig. 2", we begin with Exploratory Data Analysis (EDA) to thoroughly examine the dataset. For the tabular data, we identify and remove outliers to ensure the integrity of the analysis. Through EDA, we also determine key correlations within the data and aim to engineer valuable features using Fenet. For the time-series data, we experiment with statistical methods (mean, min, max, etc.), autoencoders, and sleep detection techniques. This results in four distinct datasets: the original tabular data, time-series data with statistical features, time-series data after autoencoder processing, and sleep-related features extracted from the time-series data. From these four datasets, we evaluate several machine learning models and apply Optuna for hyperparameter tuning. The final model incorporates an ensemble of the best-performing models, ensuring robust and accurate predictions. This methodology demonstrates both the effectiveness and efficiency of our approach in tackling the complexities of PIU prediction.

A. Data cleaning

First, we examine the tabular data using Exploratory Data Analysis (EDA). It is evident that the dataset contains numerous outliers, which contribute to data noise and adversely affect model training. For example, extreme values in certain features can distort the learning process, leading to poor generalization and inaccurate predictions. To address this, we rigorously identify and remove these outliers, ensuring that the data used for model training is cleaner and more representative of real-world scenarios. This preprocessing step not only enhances model performance but also minimizes the risk of overfitting, thereby improving the robustness and reliability of the predictions.

The time-series data in the Child Mind Institute PIU dataset requires thorough cleaning to address key issues. First, idle sleep mode, enabled in 50% of the data, introduces time gaps during periods of no motion, complicating non-wear detection and sleep analysis. Second, secondary sensors, such as light and battery voltage, are upsampled to match the 5-second resolution, creating ramping artifacts that distort the true temporal patterns. To clean the data, we identify and handle gaps from idle sleep mode by marking non-wear periods or imputing missing values, while ramping artifacts are mitigated using downsampling and smoothing techniques. These steps ensure the data is reliable and ready for feature extraction and model training.

B. Feature Engineering with tabular

Feature engineering plays a critical role in improving model performance by transforming raw data into meaningful inputs. Using Exploratory Data Analysis (EDA) combined with domain knowledge, we identified relationships between features that could be leveraged to create new, more informative features.

For example, age and physical_bmi exhibit a non-linear relationship. While younger participants generally have higher physical_bmi due to growth stages, this trend does not scale linearly with age, as physiological changes and lifestyle factors influence the BMI differently in older individuals. By combining these features, we derived new composite features, such as age-adjusted BMI or growth-factor-normalized BMI, which capture these nuanced interactions more effectively.

By combining EDA findings with domain expertise, we generated new features that integrate related variables. For example, the product of `Basic_Demos-Age` and `Physical-BMI` (`BMI_Age`) captures age-adjusted trends in BMI. Similarly, interactions between `PreInt_EduHx-computerinternet_hoursday` and `Physical-BMI` (`BMI_Internet_Hours`) reflect the relationship between internet usage and physical health. These derived features go beyond simple linear relationships, encapsulating the nuanced dynamics within the data.

FEnet uses a series of operations to combine existing features, often through different feature generation instances, such as 'FeatGen'. Common operations include multiplication ('mult') and division ('div'), which are applied to pairs of features to create new combinations. For example, multiplying two features like age and BMI might reveal how BMI is influenced by age across different groups. Similarly, dividing the body fat percentage by BMI (creating a 'Fat-to-BMI' ratio)

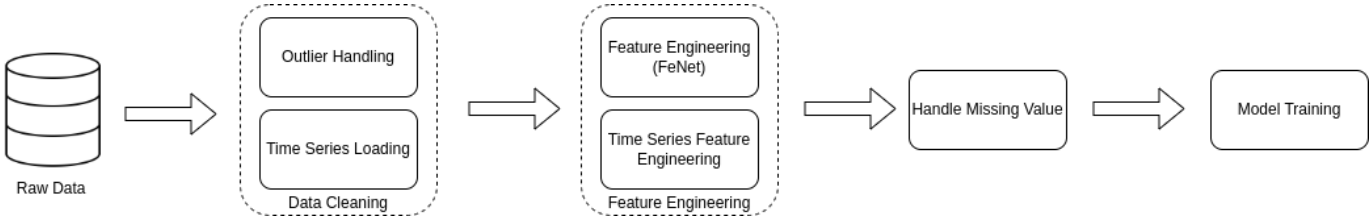

Fig. 2. Some description of the figure

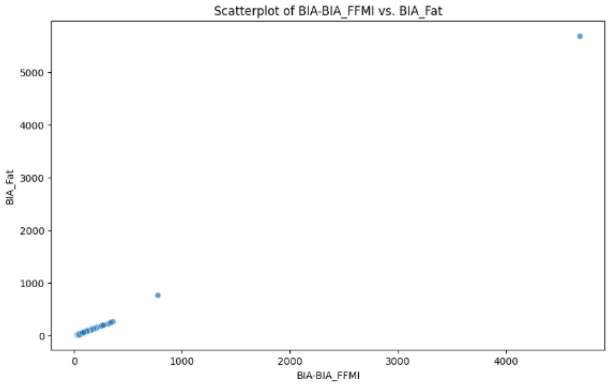

Fig. 3. Scatter plot of BIA-FFMI vs. BIA-Fat highlighting outliers. The majority of data points form a clear cluster, indicating typical relationships between BIA-FFMI and BIA-Fat values. However, several outliers deviate significantly from the main cluster, suggesting potential anomalies or data recording errors. These outliers are critical to identify and address during the cleaning process to ensure accurate analysis and modeling.

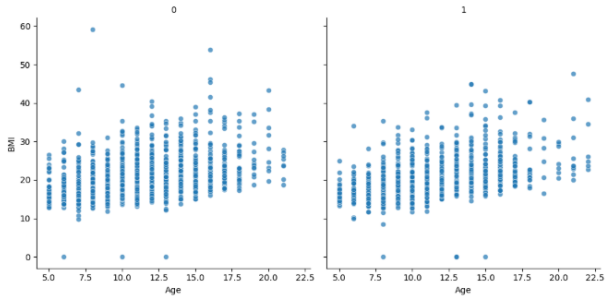

Fig. 4. Scatter plot of age vs. physical_bmi showcasing a non-linear relationship. Younger participants tend to have higher physical_bmi due to growth stages, while variations in older individuals reflect the influence of physiological changes and lifestyle factors. This insight motivates the creation of derived features, such as age-adjusted BMI, to better capture the complex interplay between these variables.

can reveal additional insights into body composition. These operations are not limited to basic arithmetic but can also include more complex transformations, such as logarithmic, exponential, or polynomial combinations.

The primary strength of FEnet lies in its ability to model feature interactions effectively. By combining features through various operations, the network captures deeper, more meaningful relationships that raw features alone may not convey. For instance, in a health-related dataset, the interaction between physical health metrics (such as BMI) and behavioral factors (such as internet usage) might reveal trends that influence predictions about mental health or internet addiction. Operations like multiplication and division help to express these interactions in a mathematically tractable form, allowing the model to learn patterns that improve its performance.

In the case of the Child Mind Institute PIU prediction task, the relationships between features can be highly complex. Features such as age, BMI, and internet usage hours might individually contribute to understanding the problem, but their combined effects are often more significant. For example, multiplying `Basic_Demos-Age` by `Physical-BMI` results in the `BMI_Age` feature, which may better capture the relationship between age and BMI than either feature alone. Similarly, the interaction between internet usage hours and BMI (`BMI_Internet_Hours`) reveals how internet use might impact physical health, which is an important factor in predicting PIU. These newly generated features encapsulate the nuanced dynamics within the data, providing the model with more robust and informative inputs.

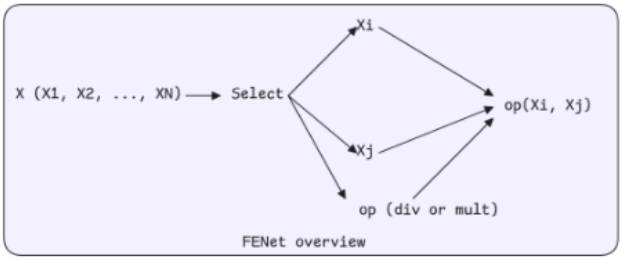

Fig. 5. Feature engineering with tabular - Fenet

C. Feature Engineering with Time Series Data

To enhance the predictive capabilities of our model, we adopted a feature engineering approach that utilizes autoencoders for time-series data. Autoencoders are a powerful type of neural network designed to learn efficient representations of input data, particularly useful when dealing with complex and high-dimensional datasets like time-series data. Let me explain why we chose this approach and how it contributes to feature engineering.

An autoencoder consists of two main components: the encoder and the decoder. The encoder compresses high-dimensional input data into a lower-dimensional latent representation, capturing the most critical features while discarding noise and redundancy. This compressed representation, often

referred to as the bottleneck or latent space, encodes the essential information required to understand the underlying patterns of the data. The decoder then reconstructs the original data from this compressed form, ensuring that no valuable information is lost during the encoding process. This architecture allows the autoencoder to capture non-linear relationships and extract meaningful features from the raw data.

During the training process, the goal of the autoencoder is to minimize the reconstruction loss, which measures the difference between the original input (X) and the reconstructed output (X') after passing through the encoder and decoder. By minimizing this loss, the autoencoder learns to extract the most important features that allow it to accurately reconstruct the input data. As a result, the features extracted through this process can provide meaningful insights into the time-series data, improving the performance of downstream models, especially in tasks where capturing temporal dependencies is critical.

The feature engineering process using autoencoders goes beyond simply generating new features; it enables the identification of the most important underlying factors that drive the observed time-series patterns. These learned features can then be used as inputs to other models, providing a more concise and informative representation of the data. By leveraging the power of autoencoders, we enhance the ability of our predictive models to capture complex, non-linear relationships within time-series data, leading to more accurate and reliable predictions in our analysis.

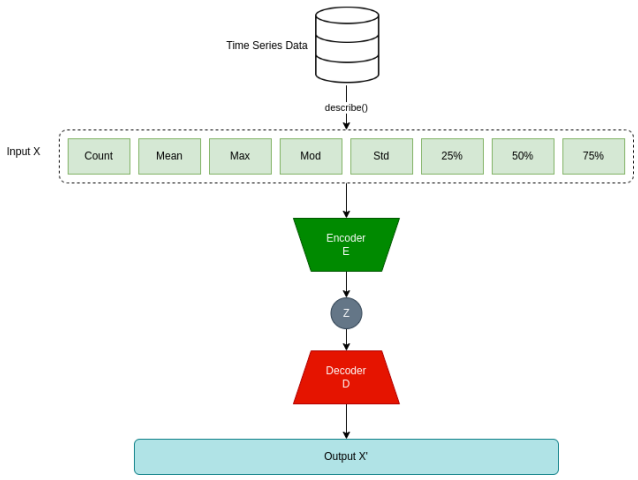

Fig. 6. Feature engineering with Time Series - Autoencoder

Next, we expanded the dataset by integrating sleep-related features derived from the Child Mind Institute's Detect Sleep States competition, which was held last year. This addition was crucial for improving the accuracy of our predictions, as research has shown that sleep patterns are closely linked to problematic internet use (PIU). Poor sleep quality or irregular sleep habits often correlate with higher internet usage, making sleep a key factor in understanding and predicting internet addiction.

By leveraging sleep-related metrics, we aimed to capture this relationship and enhance the predictive capabilities of our model. To accomplish this, we turned to the insights provided by the top solutions from last year's competition, which primarily focused on sleep state detection using actigraphy data. These solutions provided valuable methods for extracting critical sleep-related features, including:

Total Sleep Time (TST): The overall amount of sleep a person gets during the night, which can serve as an indicator of sleep quality. Sleep Efficiency: A measure of the proportion of time spent in bed that is actually spent sleeping, helping to gauge the effectiveness of sleep. Sleep Onset Latency: The time it takes to fall asleep after going to bed, which is a key measure of sleep initiation issues. By integrating these features into our model, we were able to better capture the nuanced relationship between sleep patterns and internet use, thereby improving our ability to predict PIU.

One of the most impressive solutions in the competition came from the 4th place team, led by penguin46 and their teammate. Their approach combined Neural Networks (NNs) and Gradient Boosting Decision Trees (GBDTs) into an ensemble model, incorporating thoughtful post-processing techniques to maximize performance.

The input data consisted of 19,200 steps, with the series patched every 12 steps. This effectively compressed the input time-series to a length of 1,600, making it more manageable for efficient training. As a result, the input shape was structured as (batch_size, 1600, $12 \times$ num_features), which enabled the model to process and analyze the data efficiently.

The model architecture employed several advanced techniques, including transformer layers, WaveNet, and 1D Convolutional Neural Networks (CNN), followed by a Gated Recurrent Unit (GRU). These layers allowed the model to capture both short-term and long-term temporal dependencies within the time-series data, which is essential for understanding the dynamics of sleep patterns over time.

During both training and inference, the input series was shifted by 1/8 of its length. This ensured that each position was predicted multiple times (approximately 8 times), and the final predictions were averaged for stability and robustness. This technique helped to reduce variability and improve the reliability of the model's output.

The results from this approach were impressive, with NN-based cross-validation (CV) scores ranging from 0.805 to 0.814. The predictions were further averaged over multiple seeds, which ensured greater robustness in the final output. The team also used an ensemble method to combine multiple models, achieving a higher level of accuracy and reliability in their submissions.

By integrating sleep-related features and using advanced machine learning techniques, we were able to build a more accurate and effective model for predicting problematic internet use. This approach not only enhanced the predictive power of our model but also highlighted the importance of incorporating domain knowledge—such as sleep behavior—into time-series analysis for better outcomes.

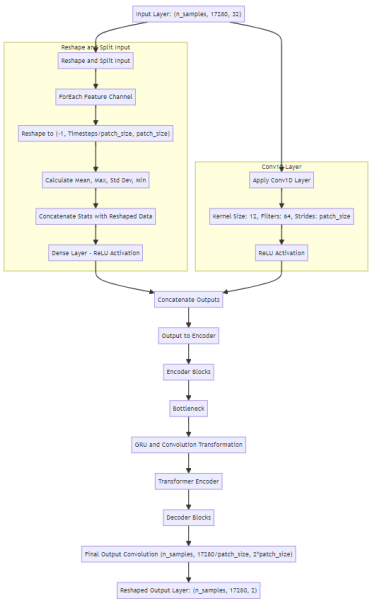

Fig. 7. Feature engineering with Time Series - Sleep Detection

D. Handle Dataset

To handle the dataset, we combined several techniques to process both time-series and tabular data. First, we applied traditional time-series preprocessing, such as smoothing and normalization, to ensure the data's consistency. We also utilized sleep detection methods to extract key features like Total Sleep Time and Sleep Efficiency, which are essential for understanding the relationship between sleep patterns and internet use. Additionally, autoencoders were used to extract meaningful features from the time-series data, capturing non-linear patterns and reducing dimensionality. To handle missing values, we employed KNN Imputation, which filled gaps in both time-series and tabular data by considering the nearest neighbors. This comprehensive approach allowed us to create a robust dataset for predictive modeling.

E. Model Training

After processing and engineering the features, we proceeded to train multiple machine learning models using the enriched dataset. To ensure the best performance, we utilized Optuna, a hyperparameter optimization framework, to fine-tune the models. Optuna automates the search for the optimal set of hyperparameters, improving model performance by exploring a wide range of possibilities for each algorithm.

For the training process, we tested various algorithms, such as Gradient Boosting, Random Forest, and Neural Networks, using the features derived from both the time-series and tabular data, including sleep-related features and autoencoded representations. With Optuna, we were able to identify the most suitable hyperparameters, such as learning rate, number of estimators, and tree depth, for each model.

Once individual models were trained and tuned, we performed model ensembling. This step involved combining the predictions from each model to create a more robust and accurate final prediction. The ensemble approach leverages the strengths of different models, reducing overfitting and improving generalization by averaging or stacking their outputs. By combining the results from various algorithms, we were able to obtain superior performance compared to any single model, ensuring a reliable prediction of problematic internet use (PIU).

IV. Experiments

To evaluate our approach, we first compare our results with the popular Kaggle notebook that achieved a score of 0.494 in a recent competition. This notebook has received significant attention due to its strong performance, where it integrates time-series data with feature extraction, preprocessing, and model training strategies. The pipeline in this notebook begins by handling missing labels and performing imputation of missing features, which are crucial for ensuring data integrity. With a focus on feature extraction from both time-series and tabular data, it employs a tree-based algorithm for prediction, showcasing its strength in handling complex datasets. This serves as a solid baseline for assessing the effectiveness of our own approach.

In our second pipeline, we follow a systematic process starting with discarding samples with missing labels, ensuring that we only use data that can meaningfully contribute to model training. Next, we impute missing features in the tabular data using techniques like KNN imputation, which takes advantage of the relationships between samples. For time-series data, we leverage autoencoders to reduce dimensionality while preserving the temporal patterns, making it easier for the model to learn from the data. After preprocessing, we apply feature selection using KBest, which helps identify the most relevant features based on their statistical significance. Finally, we train the model using three different tree-based regression algorithms, such as Random Forest, XGBoost, and Gradient Boosting, to capture both linear and non-linear patterns in the data.

Our third pipeline adapts the same preprocessing steps but applies a classification approach instead of regression. After discarding missing labels and imputing missing features, we use the same feature selection and autoencoder-based feature extraction techniques. However, this time, the goal is to predict categorical outcomes, such as classifying a subject into a specific group based on their internet use behavior. By leveraging the tree-based classifiers like Decision Trees, Random Forest, and XGBoost, we aim to identify which features best separate the different classes. This classification approach helps to understand how the features interact to classify individuals into different categories based on their sleep patterns and internet use.

Finally, in our own method, we focus on enhancing feature extraction through a combination of sleep detection, time-series data, and domain knowledge. By integrating insights from the Child Mind Institute's sleep detection competition, we extract sleep-related features such as Total Sleep Time

(TST), Sleep Efficiency, and Sleep Onset Latency from actigraphy data. These features, in combination with time-series autoencoders and tabular data features, provide a rich set of information that captures the dynamics of internet use and sleep behavior. This method aims to leverage advanced feature extraction techniques to improve the overall model's predictive performance, allowing for more accurate and robust predictions of problematic internet use.

Through these four pipelines, we aim to compare different preprocessing techniques, feature selection strategies, and model types to determine which combination of methods yields the best results in predicting problematic internet use, with a specific focus on sleep-related features and time-series data.

Seed	2024			2025		
Pipeline	CV	Private Score	Public Score	CV	Private Score	Public Score
Sleep detection	0.461	0.462	0.471	0.461	0.461	0.471
Notebook 0.494	0.501	0.406	0.494	0.487	0.399	0.454
Baseline (Regressor)	0.454	0.412	0.429	0.452	0.422	0.431
Baseline (Classification)	0.412	0.416	0.418	0.42	0.431	0.423

Fig. 8. Experiment with difference pipelines

In our comparison, it's important to note that our pipeline places a strong emphasis on data processing and feature extraction, which is why the results across different metrics, such as CV, public score, and private score, tend to be more consistent. The consistency in scores indicates that our approach is robust and generalizes well across different datasets, without overfitting to the public test set. This is in contrast to Pipeline 2 and Pipeline 3, which focus more heavily on model complexity and fine-tuning. Pipeline 2, while effective, may risk overfitting to the public test set, as it focuses on a combination of tree-based regression models, which could lead to high variance in test scores when the model is exposed to new data. Similarly, Pipeline 3, by emphasizing classification models, might suffer from a similar issue, relying too much on model-specific optimizations and thus failing to generalize effectively.

Furthermore, the Kaggle notebook scoring 0.494 raises some concerns. While it has garnered attention due to its high score, it relies on overly complex code and techniques that seem designed to overfit to the public test set rather than focusing on creating a model that can generalize across diverse data. This overfitting strategy might result in strong performance on the public test set, but it likely leads to lower performance on the private test set, as it does not account for the underlying relationships in the data in a meaningful way. In contrast, our pipeline's focus on careful data preprocessing and feature engineering allows us to generate a more robust model that performs well across both public and private test sets, reflecting its true predictive power.

Overall, the key takeaway is that while complex models and overfitting techniques might yield short-term success, they are not sustainable or reliable in the long run. Our approach prioritizes the data itself, creating features that capture important underlying patterns and ensuring that the model remains generalizable, without falling prey to overfitting. This results in more stable and consistent performance across different evaluation sets.

A. *Why we fail?*

In discussing why some pipelines fail, it's crucial to address the issue of data drift, which refers to the difference between the public test set and the private test set. This is a common challenge in machine learning competitions, where models are trained and validated on one set of data (public test) but evaluated on another (private test). The problem arises because the distribution of data in the public test set may not fully represent the broader, more diverse data seen in the private test set. As a result, models that are tuned or overfitted to the public test set can perform well on it but fail to generalize when applied to the private test set.

The discrepancies between public and private test scores are clearly illustrated by the performance of top teams in various Kaggle competitions. Some teams that score exceptionally well on the public test set find their scores drastically drop on the private test, indicating that their models have overfit to the quirks or biases present in the public test data. This overfitting is often caused by a model being too finely tuned to the specific examples in the public test set, leading to poor generalization. Conversely, teams with lower public test scores but higher private test scores may have built models that generalize better, effectively handling the nuances of the private test data.

These performance gaps between public and private test scores are a clear indication that relying heavily on public test set performance can lead to misinformed decisions during model selection. Teams that focus too much on optimizing for the public test set risk building models that are not truly robust. Instead, it's essential to prioritize generalization, focusing on techniques like feature engineering and cross-validation that ensure models are not simply memorizing the public test data but instead capturing the underlying patterns that will hold up across unseen data.

In our case, the consistent performance across both public and private tests reflects our model's ability to generalize, with a pipeline that emphasizes careful data processing, feature extraction, and avoiding overfitting strategies. This approach allows us to navigate the challenges of data drift and ensure that our model's performance is stable and reliable across various evaluation sets.

REFERENCES

[1] Z. Cai, P. Mao, Z. Wang, D. Wang, J. He, X. Fan, Associations between problematic internet use and mental health outcomes of students: A meta-analytic review, Adolescent Research Review 8 (1) (2023) 45–62.

[2] A. Restrepo, T. Scheiniger, J. Clucas, L. Alexander, G. Salum, K. Georgiades, D. Paksarian, K. Merikangas, M. Milham, Problematic internet use in children and adolescents: associations with psychiatric disorders and impairment, BMC Psychiatry 20 (2020) 1–11.

[3] S. Li, Z. Wu, Y. Zhang, M. Xu, X. Wang, X. Ma, Internet gaming disorder and aggression: A meta-analysis of teenagers and young adults, Frontiers in Public Health 11 (2023).

[4] J. Liu, S. Riesch, J. Tien, T. Lipman, J. Pinto-Martin, A. O'Sullivan, Screen media overuse and associated physical, cognitive, and emotional/behavioral outcomes in children and adolescents: An integrative review, Journal of Pediatric Health Care 36 (2) (2022) 99–109.

[5] A. Al-Amri, S. Abdulaziz, S. Bashir, M. Ahsan, T. Abualait, Effects of smartphone addiction on cognitive function and physical activity in middle-school children: a cross-sectional study, Frontiers in Psychology 14 (2023).

[6] S. Lakkunarajah, K. Adams, A. Pan, M. Liegl, M. Sadhir, A trying time: Problematic internet use (piu) and its association with depression and anxiety during the covid-19 pandemic, Child and Adolescent Psychiatry and Mental Health 16 (1) (2022) 49.

[7] Y. Yu, Y. Wu, P. Chen, H. Min, X. Sun, Associations between personality traits and problematic internet use among chinese adolescents, Journal of Adolescence (2023).

[8] K. Conyngham, J. M. Luckey, F. Pawelczyk, [proposal-ML] relating physical activity to problematic internet use, in: Submitted to Tsinghua University Course: Advanced Machine Learning, 2024, under review.
URL https://openreview.net/forum?id=DZI0v9KRU0