Relating Physical Activity to Problematic Internet Use

CS230 Project: Final Report

CA Mentor: Shijia Yang

Marcelo Fernandez	Edgar Miguel Roman
	Mustafa Abdelrahim Haroun Fadl

Introduction: Problem Overview

In today's digital era, excessive internet use among children and adolescents has become a widespread concern, raising alarms about its potential impact on young people's mental and emotional well-being. Problematic internet use, which can involve compulsive browsing, gaming, and social media engagement, is increasingly linked to negative mental health outcomes such as depression, anxiety, and social isolation. Recognizing the importance of early intervention, our project aimed to identify signs of problematic internet use at an early stage. We focused on analyzing physical activity data from users and correlating this information with the severity of their internet use patterns. By using this approach, we aimed to predict the participant's Severity Impairment Index (sii), a standard metric for measuring problematic internet use.

Dataset

The dataset for this project is available in [References]. Originally sourced from the Healthy Brain Network, a mental health study conducted in New York City.

This dataset size is about 3960 instances, and it includes physical activity metrics and fitness data from children and adolescents, along with information on their internet use. Specifically, the dataset contains two separate pillars, the first one is tabular data which contains information like the basic demographic data of the users, some physical measures, some measures associated with compulsive use of the internet like compulsivity, escapism, and dependency, and the label which is the sii. The second part is the accelerometer data, which is time series data measured through a wearable watch that the participants used for a given period of time, and it contains of physical measures (like the acceleration in x, y, and z directions) of each user through different time steps.

Preprocessing

For preprocessing, we began by removing columns unrelated to predicting the severity of internet use, such as the user ID and season-related data. Since many columns had missing values, we applied various imputation techniques. For numerical columns, we replaced missing values with the column mean, except for the age column, where we used the median. For categorical columns, we imputed missing values with the most frequently occurring value in each column. We also considered an advanced imputation method called GAIN [2], which uses a generative adversarial model with a generator and discriminator network to generate synthetic data for missing numerical values. For the label column, which had numerous missing entries, we applied a self-learning technique to

predict the missing labels.

Next, we addressed outliers by clipping extreme values within certain columns. We performed feature engineering by aggregating data from columns related to computer use into a new column that reflects overall screen time exposure.

To preprocess the accelerometer data effectively, we removed columns irrelevant to the analysis, such as battery voltage and time of day, as they did not contribute to the predictive features of interest. We filtered out rows where the wearable device was not actively worn, ensuring only meaningful and reliable data were retained. To standardize the dataset, we selected approximately 10,000 steps per user. Additionally, we excluded users who lacked accelerometer readings, narrowing the dataset to participants with both accelerometer and tabular data available for training. This preprocessing pipeline ensured that the final dataset was clean, relevant, and well-suited for training the deep learning model.

Methods and Architecture choices that were explored

To build a model capable of accurately predicting the sii, we started with simple neural network architectures. Our baseline model consisted of an input layer, a single hidden layer with 16 units, and an output layer with 4 units. We then expanded the architecture by adding a second hidden layer of 32 units and applied dropout regularization in the hidden layers with a dropout rate of 0.3. Next, we tried a wider network with 64 units in the first hidden layer and 32 units in the second. In the final architecture, we used three hidden layers with 128, 64, and 32 units respectively, applying dropout with a 0.3 rate in the first two hidden layers, along with L2 regularization (regularization parameter of 0.001) for added robustness. For activation functions, we used ReLU in the hidden layers and softmax in the output layer.
In the table below we showed the different architectures along with their final validation accuracies and losses:

Model	Layer	Number of Units	Activation Function	Regularization / Normalization	Final Accuracy / Loss
Model 0 (Baseline)	Layer 1	16	ReLU	-	71% / 0.66
	Output	4	Softmax	-	
Model 1	Layer 1	32	ReLU	Dropout (P=0.3)	70.3% / 0.62
	Layer 2	16	ReLU	Dropout (P=0.3)	
	Output	4	Softmax	-	
Model 2	Layer 1	64	ReLU	-	80.4% / 0.54
	Layer 2	32	ReLU	-	
	Output	4	Softmax	-	
Model 3	Layer 1	32	ReLU	Batch Normalization	77.2% / 0.59
	Layer 2	32	ReLU	Batch Normalization	
	Layer 3	16	ReLU	-	
	Output	4	Softmax	-	
Model 4	Layer 1	32	ReLU	L2 ($\lambda = 0.001$)	76.2% / 0.67
	Layer 2	16	ReLU	L2 ($\lambda = 0.01$)	
	Output	4	Softmax	-	
Model 5	Layer 1	64	ReLU	Dropout (P=0.3), L2 ($\lambda = 0.001$)	77.1% / 0.6
	Layer 2	32	ReLU	Dropout (P=0.3), L2 ($\lambda = 0.001$)	
	Output	4	Softmax	-	
Model 6	Layer 1	128	ReLU	Dropout (P=0.3), L2 ($\lambda = 0.001$)	83.1% / 0.51
	Layer 2	64	ReLU	Dropout (P=0.3), L2 ($\lambda = 0.001$)	
	Layer 3	32	ReLU	-	
	Output	4	Softmax	-	

For training, we employed the categorical cross-entropy loss function and the Adam optimizer with a learning rate of 0.00005. We split the input data into training and validation sets, using an 80/20 split, and applied an early stopping callback to halt training if the validation accuracy stopped improving.

In addition, we explored a model for handling both tabular and timeseries data. The proposed deep learning model was designed to process two different data sources: accelerometer time-series data and tabular static data. The architecture consisted of two independent branches that later merged to extract comprehensive features. The first branch processed the accelerometer data using a Long Short-Term Memory (LSTM) network capturing temporal dependencies and producing a 20-dimensional feature vector. The second branch employed three dense layers to process tabular data, with sizes of 128, 64, and 32 respectively. We used a dropout regularization with P=0.3 for the first two dense layers. This second branch transformed the tabular data into a 32-dimensional feature vector. The outputs of these parallel networks were then concatenated and passed through three fully connected layers with sizes of 64, 32, and 16 to refine the representation further. The final output layer produced a 4-dimensional vector that encapsulated the learned features from both modalities. The architecture of the model is shown in the following figure:

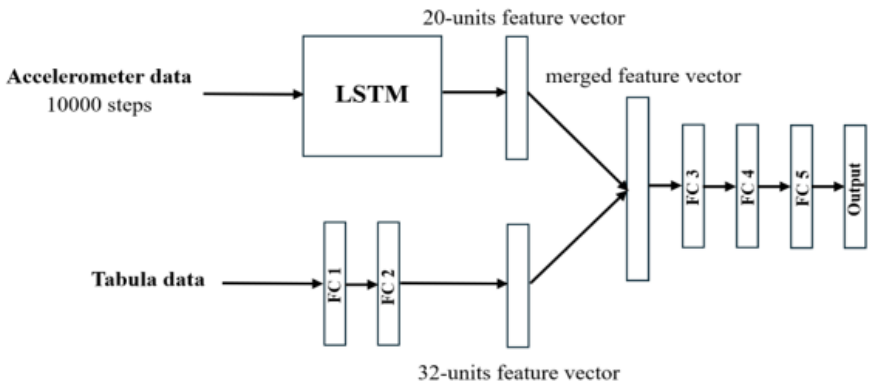

Figure 1: Architecture of the model

The training process optimized the Mean Squared Error (MSE) loss function using the Adam optimizer with a learning rate of 0.0001. The model was trained over 250 epochs with a batch size of 32. To prevent overfitting, we aplied an early stopping with a patience of 15 epochs to monitor the validation loss and restoring the best weights. The training set was split into 80% training and 20% validation data. Accuracy was tracked as an additional metric to evaluate the model's predictive performance. GPU acceleration was leveraged to handle the large input dimensions efficiently, significantly reducing the computation time required to train the model. The final validation accuracy of the model was 88.16%, and the results of training and validation losses and accuracies are shown in the following figure:

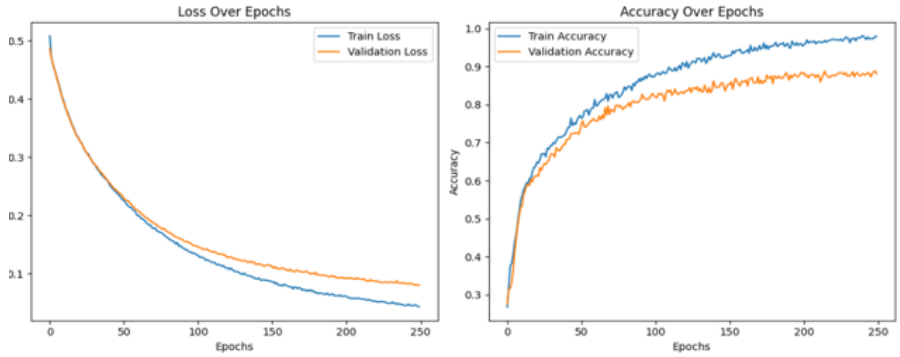

Figure 2: Results

Finally, inspired by the "Wide & Deep Learning for Recommender Systems" paper by Google[3], we incorporated a Wide & Deep neural network architecture to enhance our prediction of the Severity Impairment Index (sii) using tabular data. This model combines a wide (linear) component and a deep (neural network) component, enabling it to learn both low-level feature interactions and high-level abstractions from the data.

Results and Analysis

The deep learning models exhibited varying performance, highlighting the trade-offs between complexity and accuracy:

- Model 6: Achieved 83.1% validation accuracy and 0.51 loss, balancing overfitting and generalization through its deeper architecture, dropout, and L2 regularization.
- Baseline Model (Model 0): Served as a comparison, achieving 71% accuracy and demonstrating the improvements gained with more advanced architectures.
- Combined Tabular and Time-Series Model: Achieved the highest validation accuracy of 88.16% by leveraging an LSTM branch for accelerometer data and a dense branch for tabular data, effectively capturing temporal and static dependencies.
- Wide & Deep Model: Improved generalization over traditional neural networks by combining L1 regularization for direct feature interactions and L2 regularization for complex pattern learning.

Training curves showed stable convergence, aided by hyperparameter tuning, regularization, and early stopping to prevent overfitting. These findings demonstrate the effectiveness of integrating physical activity data and advanced deep learning techniques to predict problematic internet use in children and adolescents.

Insights and discussions relevant to the project

We strongly believed that this project underscored the growing potential of deep learning to address societal challenges, such as problematic internet use among children and adolescents. Some insights:

- Multimodal Data Integration: Combining time-series accelerometer data with static tabular data was critical for capturing behavioral patterns and contextual factors effectively
- Wide & Deep Model: The model highlighted the strength of integrating linear and non-linear modeling approaches. The wide component efficiently captured direct feature interactions, while the deep component modeled complex non-linear relationships, resulting in a robust predictive framework. Insights from recommender systems were successfully adapted to health informatics, emphasizing the importance of tailoring architectures to data characteristics and problem specifics.
- Domain-Specific Architectures: Using LSTMs for sequential data and fully connected layers for static features demonstrated the value of aligning model design with data nature. Effective preprocessing, including data imputation, feature standardization, and advanced techniques like GAIN, was crucial for model performance, offering avenues for further improvement.
- Regularization and Overfitting: Regularization techniques, including dropout and L2 penalties, were pivotal in enhancing generalization, particularly for small or imbalanced datasets, mitigating overfitting, and ensuring robust performance.
- Future Directions: Exploring real-world applications in clinical and educational settings and addressing ethical challenges can extend the project's impact and applicability.

Contributions

Marcelo, Edgar and Mustafa worked together as a team on the following items:

- Data Preprocessing: Developed the initial data preprocessing pipeline, handling missing values through imputation, encoding categorical variables, and standardizing numerical features to prepare the raw data for modeling, ensuring it was clean and suitable for training.
- Model Development Framework: Set up a structured framework for model development using tabular data, defining various neural network architectures—including the Wide & Deep learning model inspired by the "Wide & Deep Learning for Recommender Systems" paper—and implementing training routines with validation and early stopping.
- Performance Visualization: Incorporated visualization tools to monitor performance metrics like loss and accuracy over epochs, enabling the team to efficiently experiment with different models, validate their performance, and visualize results, thus facilitating collaborative development and iterative analysis.
- Preprocessing Accelerometer Data: Developed and implemented a preprocessing pipeline for accelerometer time-series data, including cleaning, filtering, and standardizing data to ensure reliability and relevance for deep learning analysis.
- Multimodal LSTM Model Development: Designed, built, and trained a multimodal deep learning model integrating LSTM networks for accelerometer data and dense layers for tabular data

Code

Github Repository

References

1 Healthy Brain Network, "The Healthy Brain Network (HBN) Data Set," Kaggle, 2024. [Online]. Available in https://www.kaggle.com/competitions/child-mind-institute-problematicinternet-use/data.

2 J. Yoon, J. Jordon, and M. van der Schaar, "GAIN: Missing Data Imputation using Generative Adversarial Nets," Proc. 35th Int. Conf. Mach. Learn. (ICML), Stockholm, Sweden, 2018, pp. 1-10.

3 J. Cheng, H. et al. (2016). Wide & deep learning for recommender systems. Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, 7-10.