Simon Kuznets Kharkiv National University of Economics

Serhii Minukhin

Oleksandr Semenets

Intelligent system for diagnosis and personalized therapy prediction in diabetes using machine learning models

Main goal:

Develop an AI model that, based on clinical and lab data, automatically predicts daily insulin and oral drug doses for diabetes patients.

Pipeline:

1. Diabetes screening: yes or no
2. Diabetes type classification: T1DM or T2DM
3. Therapy prediction: daily insulin and (or) tablets

Machine Learning Methods

Traditional models: Decision Tree, Random Forest, Logistic Regression, KNN

Boosting models: XGBoost, LightGBM

Deep learning models: GRU, LSTM

Pipeline

Stage 1

Data → Preprocessing (improved missing values, normalization, summaries) → Models (Random Forest, Decision Tree, KNN, Logistic Regression, XGB, LGBM) → Evaluation → Top Models: LGBM, XGB, Random Forest

Stage 2

Data → Preprocessing → Models (same as Stage 1) → Evaluation → Top Models: LGBM, XGB, Logistic Regression

Stage 3

Data → Preprocessing → Database → Feature Engineering (handle static and dynamic data) → Model Training → Evaluation → Predictions (drug type and daily dosage (drug–dose tensor))

Metric for 1-st and 2-nd stages

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

TP (True Positive) - correctly predicted positive cases.
TN (True Negative) - correctly predicted negative cases.
FP (False Positive) - the model incorrectly predicted “positive” when it was actually “negative.”
FN (False Negative) - the model incorrectly predicted “negative” when it was actually “positive.”

Results of the 1-st stage metrics

Random Forest Classifier	**0.9498**
Decision Tree Classifier	0.9405
KNeighbors Classifier	0.9247
Logistic Regression	0.9286
XGB Classifier	**0.9693**
LGBM Classifier	**0.9708**

Results of the 2-nd stage metrics

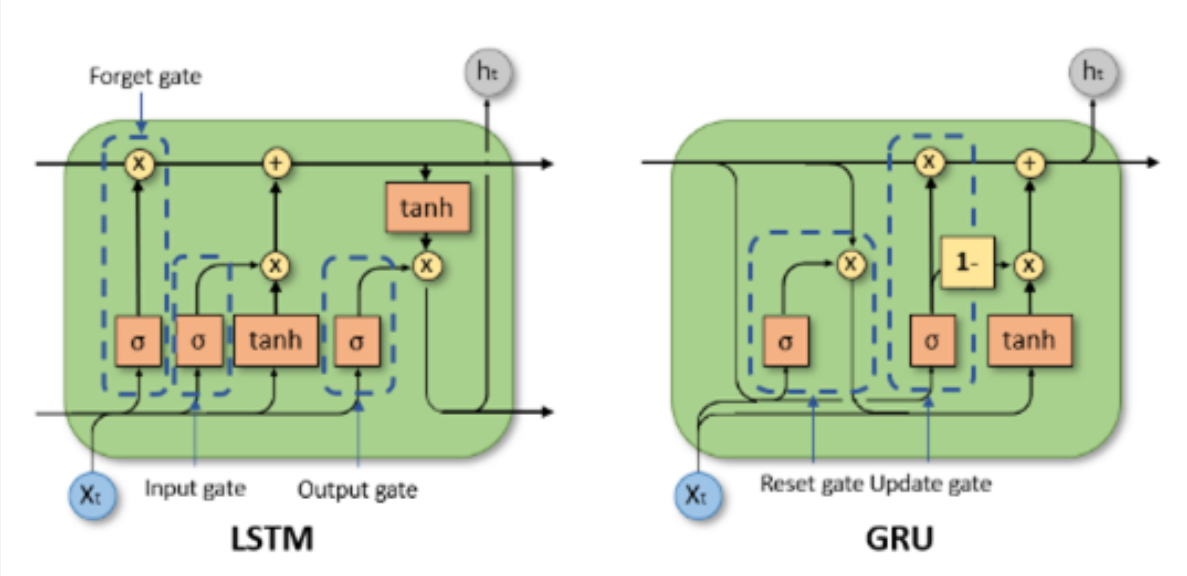
Random Forest Classifier	**0.952**
Decision Tree Classifier	0.801
KNeighbors Classifier	0.884
Logistic Regression	0.956
XGB Classifier	**0.963**
LGBM Classifier	**0.965**

3-rd main stage

Goal: Develop a model using **GRU and LSTM** to predict both drug type and daily dosage from patient clinical and temporal data, outputting a tensor linking each drug to its predicted dose.

Database: PostgreSQL. **ORM:** SQLAlchemy

What are GRU and LSTM?



Data Preparing:

Load and merge patient data
Process insulin and non-insulin agents
Handle missing values and clinical summaries
Create a normalized dataset

Database / SQLAlchemy:

Load data using SQLAlchemy
Manage transactions: import, commit, rollback
Repository setup for model data access

Model Training:

Static data: feature selection, normalization, categorical processing
Dynamic data: patient grouping, temporal windows, PyTorch Dataset
Combine static and dynamic features
Train LSTM and GRU models for 10 epochs
Evaluation metric: Mean Squared Error (MSE)

Results of the 3rd stage metrics (MSE)

LSTM (Long Short-Term Memory): A special kind of neural network that can remember important information over long sequences of data. It's great for time-dependent data like patient history, stock prices, or speech.

GRU (Gated Recurrent Unit): Similar to LSTM but simpler and faster. It also remembers important information over sequences, but with fewer steps, making it quicker to train.

GRU (10 epoch)	LSTM (10 epoch)
0.019016	0.015577
0.007194	0.007005
0.006796	0.006735
0.006456	0.006561
0.006213	0.006453
0.006026	0.006241
0.005748	0.006261
0.005561	0.005922
0.005375	0.005735
0.005205	0.005603

Conclusions

1. Gradient boosting models (LGBM, XGB) consistently achieved the best accuracy.
2. Decision Tree and KNN showed weak performance across both stages.
3. GRU and LSTM effectively captured temporal patterns, with steadily decreasing MSE.
4. Combining boosting (for tabular data) and recurrent networks (for sequences) enables accurate prediction of drug type and daily dosage.

Sources:

1. https://www.kaggle.com/code/tumpanjawat/diabetes-eda-random-forest-hp
2. https://www.sqlalchemy.org/
3. https://figshare.com/articles/dataset/Diabetes_Datasets-ShanghaiT1DM_and_ShanghaiT2DM/20444397
4. https://docs.pytorch.org/docs

Thank for Your attention!