**Recurrent NN model for glycemic state prediction**

**Glycemic state prediction on a single patient data - LSTM model with Attention**

Following the limited success of simpler models, trying to predict the glycemic states/values, we have been working on implication of LSTM model (recurrent neural network models, designed to learn and remember patterns in sequences over long time intervals, avoiding the vanishing gradient problem)

Models of this type can control the flow of information, maintain long-term dependencies, and can be used as a regressor or a classifier for time-series data.

In our case, we have chosen an LSTM model with advanced base architecture, including bidirectional core layers, attention mechanisms, regularization, and dropout.

The model input is a set of 30-minute sequences of features extracted from 2-minute windows with 50% overlap (including insulin-related features) to predict glycemic events, happening with a 15-minute lag.

The initial idea was to train the model on some base patient data and then freeze the core layers for transfer learning and fine-tuning on other patients. However, this approach does not seem to be working well (model does not generalize, limited dataset)

The likely reason is either physiological differences between patients - such as variations in metabolism or an initially weak assumption concerning EMF-Glucose correlations.

“Individual” models have been trained for each patient / channel separately, using semi time-aware train–validation–test splits. (otherwise, consequent splitting causes strict class imbalance – no train data on hypoglycemia for most of the patients)

**Feature Preprocessing**- Converting categorical glucose states ('hypoglycemia', 'normal', 'hyperglycemia') to numeric targets (0, 1, 2)  
- Application ofrobust scaling to handle outliers in feature data

**Sequence Processing**  
- Time-ordered windows (30min each) of features with configurable stride  
- Each sequence predicts the future glycemic state target (hypo/normal/hyperglycemia)

**Class Imbalance Handling**  
- The pipeline calculates class weights based on sample distribution  
- Applies weighted loss to prioritize minority classes (in our case it is typically hypoglycemia)  
- Ensures balanced evaluation metrics through stratified sampling  
  
**Model Architecture** (basic, no fine tuning of architecture or other meta-parameters, except learning rates)  
- Bidirectional LSTM Core: Two stacked BiLSTM layers (64→32 units)  
- Attention Mechanism: Multi-head self-attention (2 heads) with residual connections for temporal pattern recognition (to improve gradient flow during training)  
- Parallel Processing: Convolutional branch (32 filters) captures local patterns  
- Feature Fusion: Concatenation of attention and convolutional outputs  
- Dual Pooling: Both global average and max pooling to preserve different statistical properties  
- Output Layer: Dense layer with softmax activation for 3-class classification  
  
**Regularization Techniques**  
- Progressive dropout (15%->30%)  
- L2 regularization on weights  
- Batch normalization after each major layer  
- Residual connections for stable gradient flow  
- Includes learning rate scheduling for optimization stability  
  
**Optimization Strategy**  
- Adam optimizer with exponential learning rate decay  
- Sparse categorical cross-entropy loss function

**Evaluation Framework**  
- Supports both standard train/validation/test splits and k-fold cross-validation (not implemented)  
- Calculates detailed metrics

**Transfer Learning Capabilities** (optional, results unsuccessful)  
- This specific approach can first patient as base model to establish foundational patterns  
- Freezes early layers to preserve learned features  
- Fine-tunes later layers for patient-specific adaptation  
- Enables effective learning even with limited per-patient data  
  
While this architecture is not fine tuned, it still effectively balances feature extraction power with regularization to predict future glycemic states from temporal magnetic signal patterns.

* It is important to mention that the model is fed with time-sorted 30min sequences of overlapping windows (extracted features), and 15 min lagged predictive state (target). These values can be tuned to give us better metrics.
* The Insulin Value (model-based) has been included into the inputs. The importance of this feature is not validated.
* The Train-Test-Validation spit is 70% / 15% /15% over the entire experiment time, but the time-aware sequences in the dataset have been shuffled inside these splits (potential for a data-leak effect). It can be considered not a classical data-leakage, if the assumption is that the 30min sequence length captures all necessary temporal context is correct and we are learning the general sequence-to-label mappings irrespective of their global chronological position (segment stationarity). In addition to that, shuffling potentially makes better regularization
* Unfortunately, since all the experiments are very repetitive (Hypo-Normal-Hyper), strict time split will lead to overfitting to this Macro-Sequence with class imbalance potential.
* Some of the Features, extracted from the signal potentially autocorrelative and might be removed in the process of model fine-tuning.

**Results**

On the first look, the figures of predicted states below look “good” in terms of individual accuracy for each patient, however important to note that recurrent models with attention are extremely strong predictors and can virtually learn the entire signal.   
The attempt to run cross-inference (study on one patient and predict the other) has failed in most of the cases. Either there is a significant inter-personal variability of the EMF signal, or the model is based on weak assumptions

**Individual (“good”) results**

* On the charts below one can see the glucose/insulin graph and predicted state (hypo/normal/hyper) colored by blue/green/red respectively. Color depth represents the confidence of the prediction.
* Over the background there is a raw signal for the specific patient/channel
* The initial part of the signal is uncolored, since it is a learning sequence for the lagged recurrent model.

A screenshot of a graph

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

A screenshot of a graph

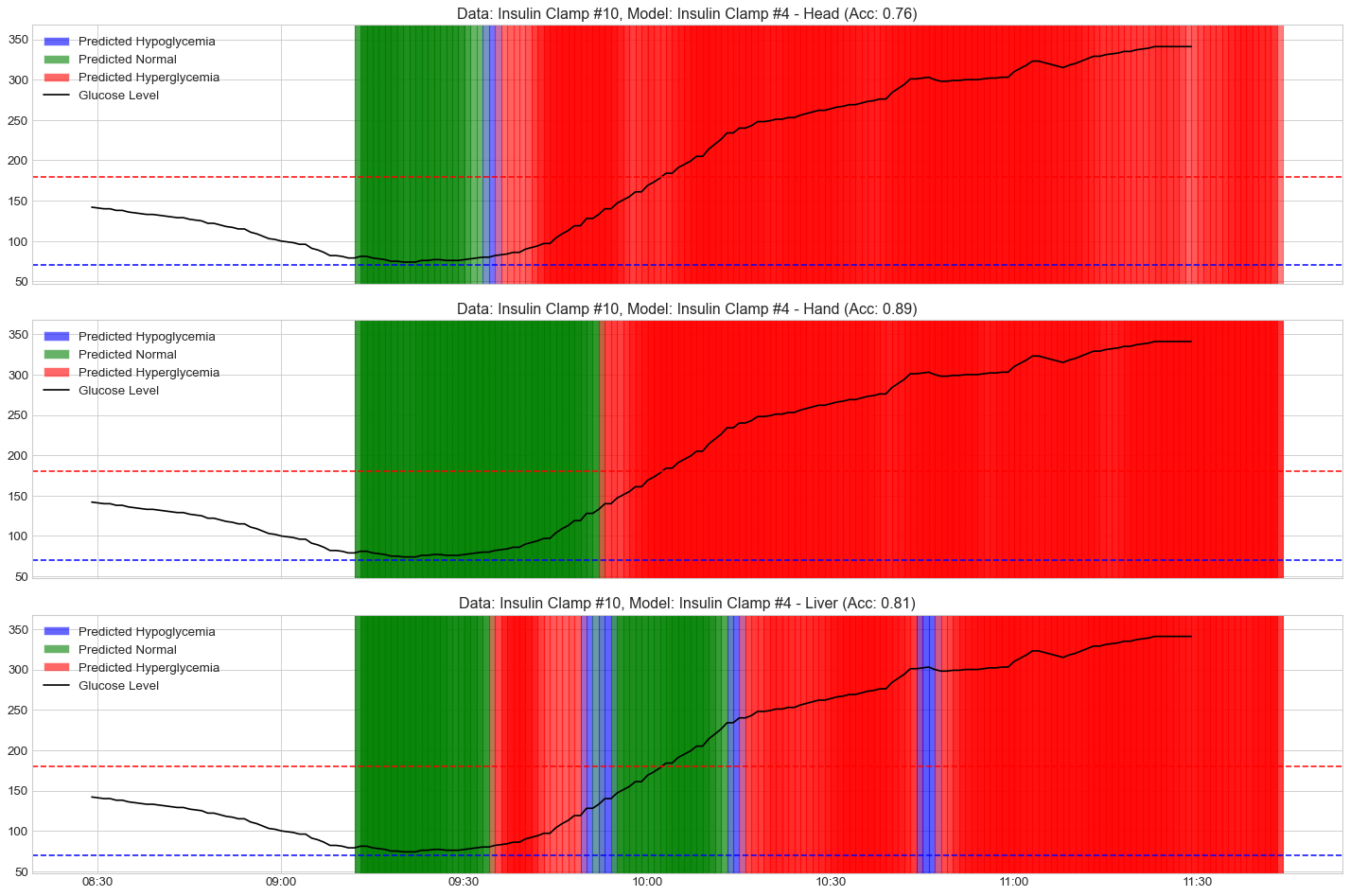
AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

While these results look acceptable, at least at the scale of state-predictivity, cross-validation across patients does not yield satisfactory results, as one may see on the figures below.

When a model trained on patient X is used to infer outcomes for patient Y, performance drops significantly. This indicates **poor generalization across individuals**. While the model may be able to detect hyperglycemic states within the training patient, it fails to transfer that capability to others - suggesting it's capturing idiosyncratic rather than universal features.

On the figures below (part of all 90 cross-validation charts), each chart represents inference on patient X, of the model, trained in patient Y. One can see that models do not generalize. Relatively high accuracy in some of the cases is related to strong class imbalances (most data is hyperglycemia) A screenshot of a graph

AI-generated content may be incorrect.

**Binary classification of glycemic states 10 minutes in advance**

After the failure of strong recurrent model (LSTM) to be generalized over all patients, we have tried another approach with a simpler model architecture for binary classification of the glycemic states (Hyper/No-Hyper), including a leave-one-out training on a group of patients. This is the simplest predictive model that might be valid in terms of medical interest.

**Data Processing Pipeline**  
  
1. Data Loading: Loading preprocessed/filtered patient sensor data (from parquet files) and glucose measurements with corresponding labels  
2. Feature Extraction: Extracting time-domain and frequency-domain features from EMF data  
3. Preprocessing: Standardizing features and creates sequential datasets for LSTM models  
4. Model Training: Implementing leave-one-patient-out cross-validation  
5. Evaluation: Calculating performance metrics (accuracy, sensitivity, specificity, AUC)  
  
**Model Architecture**  
  
The core model is a relatively simpler sequential LSTM network with attention mechanism:  
- Input layer matching feature dimensions  
- Multiple LSTM layers (default: 64, 32 units) with dropout  
- Attention mechanism to focus on relevant parts of the sequence  
- Dense output layer with sigmoid activation (binary classification)  
  
Input -> LSTM(64) -> Dropout -> LSTM(32) -> Attention -> Dense -> Output (binary class)  
  
**Data Splitting Strategy**  
This approach, unlike the previous one, uses a leave-one-patient-out cross-validation approach:  
- Training set: Data from N-1 patients (from the selected group of patients)  
- Test set: Data from the held-out patient (from the selected group of patients)  
- Sequence creation: Windows of sensor data with configurable sequence length  
- Each window is associated with a binary label (hyperglycemia/non-hyperglycemia)  
- Pre-set groups of patients by weight/age (young, old, shallow, etc.)

Example:  
- "Young\_Patients": ["Insulin Clamp #4", "Insulin Clamp #7", "Insulin Clamp #8", "Insulin Clamp #9"],  
- "Shallow": ["Insulin Clamp #3", "Insulin Clamp #6", "Insulin Clamp #7", "Insulin Clamp #9", "Insulin Clamp #10"]

**Results**

Processing group: **Young Patient**s for hyperglycemia prediction

Channel: **Head**, Glucose lag: 10 minutes

Patients: ['Insulin Clamp #4', 'Insulin Clamp #7', 'Insulin Clamp #8', 'Insulin Clamp #9']

Group **Young Patients** Average Metrics:

Accuracy: 0.5340, AUC: 0.6146, F1: 0.5234

Sensitivity: 0.5052, Specificity: 0.6402

Processing group: **Shallow** for hyperglycemia prediction

Channel: **Head**, Glucose lag: 10 minutes

Patients: ['Insulin Clamp #3', 'Insulin Clamp #6', 'Insulin Clamp #7', 'Insulin Clamp #9', 'Insulin Clamp #10']

Group Shallow Average Metrics:

Accuracy: 0.3756, AUC: 0.4532, F1: 0.3366

Sensitivity: 0.4057, Specificity: 0.5388

Processing group: **Young Patients** for hyperglycemia prediction

Channel: **Hand**, Glucose lag: 10 minutes

Patients: ['Insulin Clamp #4', 'Insulin Clamp #7', 'Insulin Clamp #8', 'Insulin Clamp #9']

Group **Young Patients** Average Metrics:

Accuracy: 0.5230, AUC: 0.5973, F1: 0.4880

Sensitivity: 0.4225, Specificity: 0.6664

Processing group: **Shallow** for hyperglycemia prediction

Channel: **Hand**, Glucose lag: 10 minutes

Patients: ['Insulin Clamp #3', 'Insulin Clamp #6', 'Insulin Clamp #7', 'Insulin Clamp #9', 'Insulin Clamp #10']

Group Shallow Average Metrics:

Accuracy: 0.5917, AUC: 0.7130, F1: 0.4331

Sensitivity: 0.4029, Specificity: 0.7969

Processing group: **Young Patients** for hyperglycemia prediction

Channel: **Liver**, Glucose lag: 10 minutes

Patients: ['Insulin Clamp #4', 'Insulin Clamp #7', 'Insulin Clamp #8', 'Insulin Clamp #9']

Group Young Patients Average Metrics:

Accuracy: 0.6810, AUC: 0.6797, F1: 0.5882

Sensitivity: 0.5515, Specificity: 0.8724

Processing group: **Shallow** for hyperglycemia prediction

Channel: **Liver**, Glucose lag: 10 minutes

Patients: ['Insulin Clamp #3', 'Insulin Clamp #6', 'Insulin Clamp #7', 'Insulin Clamp #9', 'Insulin Clamp #10']

Group Shallow Average Metrics:

Accuracy: 0.8068, AUC: 0.8881, F1: 0.7805

Sensitivity: 0.7500, Specificity: 0.8609

In most cases, the metrics are below average (F1 score of 0.5 on binary classification is a sign of random result). Somewhat better results on Shallow patient group with Liver channel might be due to local class imbalances.

**Summary and Conclusion:**

Given that LSTM is a powerful function approximator, it can eventually be overfit to individual patterns - which may lead to over-optimistic performance that doesn’t generalize to unseen patients.

Assuming the model of this type is valid in principle - that is, it truly captures EMF signals related to glycemic events, and that these events can be injectively mapped to blood glucose levels (a non-trivial and as yet unproven assumption) - then the system likely reflects highly personalized, individual hidden Markovian state dynamics.

In this case, alongside the personalized model training, the training itself could emphasize patient-specific aspects such as the appropriate lag times, input window lengths, and channel relevance. For example, a 15-minute lag might suffice for predicting hyperglycemia, whereas hypoglycemia could require a longer history window depending on the patient.

It’s unclear whether common, generalizable predictors can be reliably identified from the signal features alone. With some caution, based on the previous findings, we can say that signal activity, mobility, and complexity metrics appear to differ across metabolic states - for example, before and after glucose or insulin administration. This is evident in dimensionality reduction methods like LDA.

However, these features do not directly predict glycemic states. If reliable predictors existed and were simple, we would likely have detected them already using simpler models. Instead, any such predictors - if they exist - are likely nonlinear, complex, and highly individual.

That said, if we assume that consistent signal dynamics exist within the same individual across energetic states, it becomes feasible to train a personalized predictive model.

For example, if CGM and Insulin data are recorded over an extended period for a specific person, alongside EMF signals, and if their metabolic response remains relatively stable, then a fine-tuned model could (potentially) capture this person's specific glycemic dynamics.

Such a model may need to incorporate contextual individual features like movement, sleep–wake cycles, stress, or circadian patterns, to fully capture variability in metabolic response.

**Appendix - Extracted Features List**

**Window Metadata Features**

|  |  |
| --- | --- |
| Feature | Description |
| window\_start | Starting timestamp of analysis window |
| window\_end | Ending timestamp of analysis window |
| window\_center | Middle timestamp of analysis window |

**Time Domain Features (per channel)**

|  |  |
| --- | --- |
| Feature | Description |
| {channel}\_mean | Mean signal value |
| {channel}\_std | Standard deviation of signal |
| {channel}\_min | Minimum signal value |
| {channel}\_max | Maximum signal value |
| {channel}\_range | Range of signal values (max-min) |
| {channel}\_kurtosis | Kurtosis of signal distribution |
| {channel}\_skewness | Skewness of signal distribution |

**Signal Analysis Features**

|  |  |
| --- | --- |
| Feature | Description |
| {channel}\_zero\_crossings | Number of times signal crosses its mean |
| {channel}\_mean\_abs\_amplitude | Mean absolute amplitude relative to signal mean |

**Frequency Domain Features**

|  |  |
| --- | --- |
| Feature | Description |
| {channel}\_{band\_name}\_rel\_power | Relative power in specific frequency band |

Where band\_name includes bands from delta1\_1 (0.1–0.3 Hz) through gamma2 (35.0–40.0 Hz)

**Spectral Features**

|  |  |
| --- | --- |
| Feature | Description |
| {channel}\_spectral\_centroid | Weighted mean of frequencies in signal |
| {channel}\_spectral\_entropy | Entropy of power spectral density |

**Wavelet Features**

|  |  |
| --- | --- |
| Feature | Description |
| {channel}\_wavelet\_level\_{j}\_energy | Energy in wavelet decomposition level j (0-5) |

**Derivative Features**

|  |  |
| --- | --- |
| Feature | Description |
| {channel}\_mean\_derivative | Mean rate of change of signal |
| {channel}\_max\_abs\_derivative | Maximum absolute rate of change |
| {channel}\_mean\_acceleration | Mean second derivative (acceleration) |
| {channel}\_max\_abs\_acceleration | Maximum absolute acceleration |

**Hjorth Parameters**

|  |  |
| --- | --- |
| Feature | Description |
| {channel}\_hjorth\_mobility | Square root of variance of first derivative divided by variance of signal |
| {channel}\_hjorth\_complexity | Ratio of mobility of first derivative to mobility of signal |

**Entropy Measures**

|  |  |
| --- | --- |
| Feature | Description |
| {channel}\_sample\_entropy | Measure of signal complexity/regularity |
| {channel}\_perm\_entropy | Complexity in signal ordering patterns |
| {channel}\_app\_entropy | Approximate entropy of signal |

**Trend Analysis**

|  |  |
| --- | --- |
| Feature | Description |
| {channel}\_linear\_trend\_slope | Slope of linear regression fit to signal |
| {channel}\_linear\_trend\_r\_squared | R-squared value of linear fit |

**Inter-window Dynamic Features**

|  |  |
| --- | --- |
| Feature | Description |
| {channel}\_mean\_change | Change in mean from previous window |
| {channel}\_std\_change | Change in standard deviation |
| {channel}\_spectral\_centroid\_change | Change in spectral centroid |
| {channel}\_hjorth\_mobility\_change | Change in Hjorth mobility |

**Target Variables**

|  |  |
| --- | --- |
| Target | Description |
| glucose\_mean | Mean glucose value in current window |
| future\_glucose\_mean | Mean glucose value in future window (prediction target) |
| glucose\_category | Current glucose category (hypoglycemia, normal, hyperglycemia) |
| future\_glucose\_category | Future glucose category (prediction target) |
| metabolic\_state | Current metabolic state (Fasting, First Insulin, Ensure, Second Insulin) |
| Insulin | Current insulin level |
| glucose\_trend\_mg\_dl\_min | Rate of glucose change (mg/dL/min) |
| glucose\_trend\_cat | Categorical glucose trend (rising, slowly rising, steady, slowly falling, falling) |