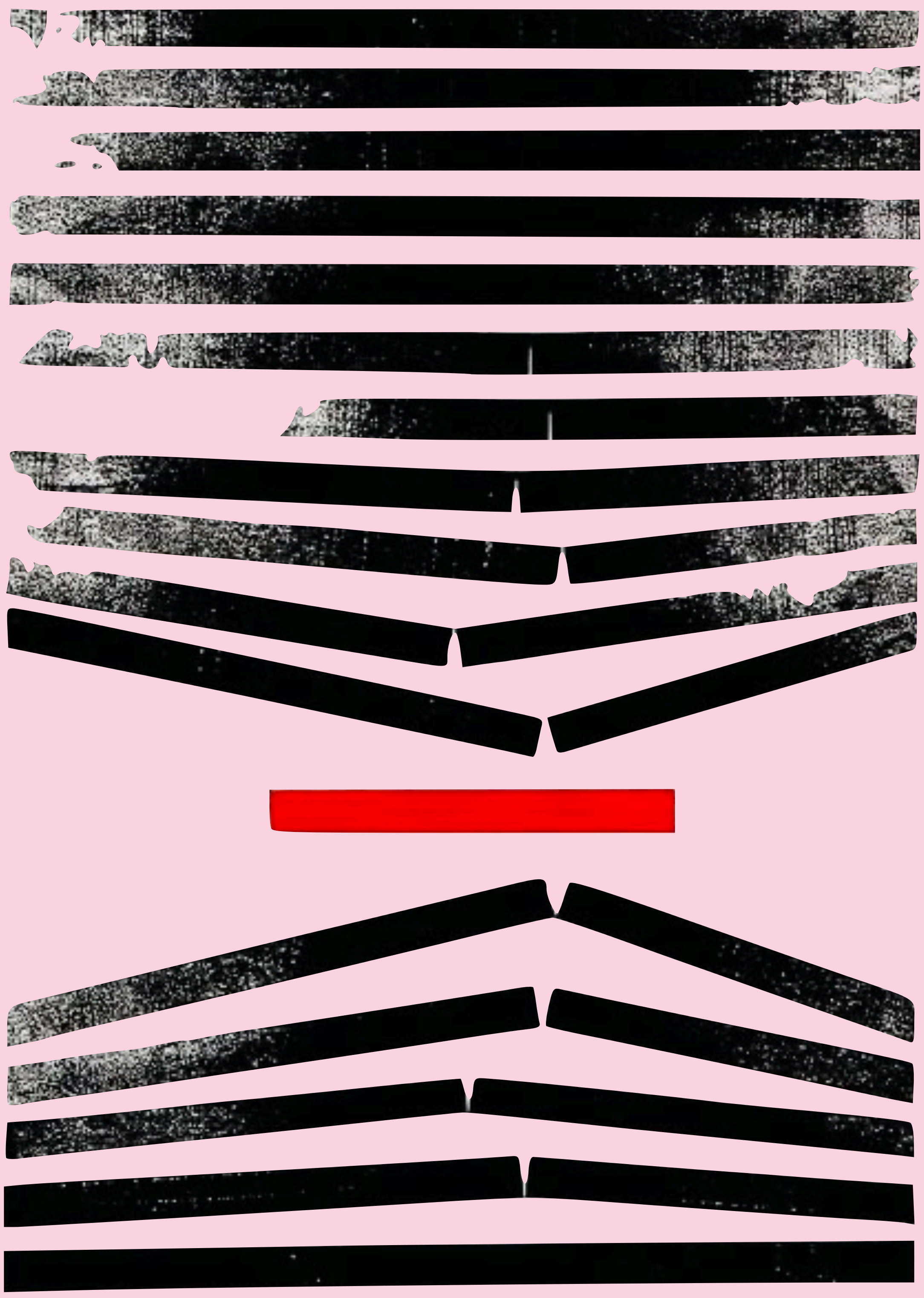
Explainable Self-Adaptive Forecasting

A tool for
interpreting | unlocking
the inner workings of
time-series forecasting



Paul Barna
Senior ML Scientist
Narrative

Model interpretability

Explaining feature importance and accumulated local effects.

The first task is to generate a pipeline to explain feature importance and accumulated local effects, using SHAP, while enabling a set of visualisations abstracted from classical notebooks, but instead hosting a dashboard which retrieves and displays the information in real time.

With a need for using a surrogate model to approximate the predictions of the underlying neural model as accurately as possible and to be interpretable at the same time, the tree-based modules with the popular variant of gradient boosted trees is often the right choice to compute the full SHAP values. This would enable a comprehensive set of analysis, including global interpretation methods like feature importance, feature dependence, interactions, clustering and summary plots or local interpretability where each feature value is a ‘force’ that either increases or decreases the prediction.

By capturing tensors while training the core neural model (incl. gradients and weights distributions), a complete understanding is achieved on how the network performs with further insights into how to tune it accordingly.

Self-Adaptive Covariates

A design of embedded self-attention signals which learn accumulated local effects which grow over time, group level effects, as well as significantly improve the predictor's best realisation.

A novel approach for use cases where:

- accumulated local effects [which grow over time] shall not be disregarded or attributed to noise
- non-stationarity is a known factor [explicitly modelled as part of the null hypothesis]
- group-level effects shall not be lost within the overall predictive distribution
- missing data [large gaps] shall not compromise the learning of a progressive underlying process
- a predictor's best realisation [the univariately built multi-step projection] shall not be compromised by the overall predictive distribution

Neural Forecasting

A state-of-the-art Autoregressive Recurrent Network for multi-step forecasts of multiple-related time series. [DeepAR].

A neural approach for use cases where:

- interested in the full predictive distribution, not just the single best realisation
- there is significant uncertainty growth over time from the data
- widely-varying scales which requires rescaling and velocity-based sampling

Probabilistic Estimates

Estimating the probability distribution of a time series' future given its past.

This is achieved in the form of Monte Carlo samples, by computing quantile estimates for all sub-ranges in the prediction horizon.

What & How It's Done

What

Explicit modelling is required to overcome the challenges of missing data, while preserving the integrity of the signal, and to maintain the predictor's best realisation (a univariately built multi-step projection).

How

The proposal is using either spline interpolation, followed by a triple exponential smoothing (where the smoothing parameters are optimised for each series independently) or using Neural Prophet (A Neural Network based Time-Series model), both to generate a synthetic pair for each time-series. Additionally, the selected after-mentioned univariately built model is used to predict n steps ahead (n - prediction length) and thus prepare the series best realisation replica, which is used down the line as a covariate for each target.

What

Accumulated local effects which are deviant from the general data distribution could uncover a temporary or permanent change within the underlying processes.

How

Proposing a covariates-aided modelling which learns where the deviation has occurred as well as what has caused it, as a integrative part of the core neural forecasting setting. Outliers are detected using a constructed empirical copula to predict tail probabilities for each data point. Quantifying the abnormality contribution of each dimension within a designated covariate for each predictor, will allow the model to focus on relevant subspaces interactions. Additionally, a global "extremeness" score is computed with an Auto-Encoder setting, followed by a heuristic contamination threshold and a user-defined rolling window. This is translated into a covariate variable shared between the predictors with the main objective of conveying data drift.

What

Allowing the model to learn item-specific behaviour or group-level effects would significantly improve the overall estimates distribution.

How

Grouping the time-series both by a user-specified category, as well as by the cluster it belongs to, after running a K-means time series clustering with dynamic time warping.