Stochastic prediction for heavy-tailed processes using extreme gradient boosting

Jan Szymczak and Paweł D. Domański

***Abstract*— This research proposes novel solution to important control engineering issue to predict stochastic properties of uncertain systems. Frequently such systems are subject to strange uncertainties with resulting time series subject to frequent outliers. Therefore, any associated predicting approach should efficiently operate with heavy-tailed data. There exist modeling tools that incorporate artificial intelligence methods based on the XGBoostLSS framework in the task of stochastic prediction. The solution proposed in the paper extends known LightGBMLSS methodology with heavy-tailed distributions having four degrees of freedom: α-stable for two-sided data and four-parameter kappa in case of one-sided data and extreme value analysis. Therefore, it allows to effectively model disturbances occurring in different research areas. Proposed approach is validated with process industry data.**

***Index Terms*— stochastic prediction; machine learning; XGBoostLSS; gradient boosting; heavy tails; α-stable distribution; four-parameter kappa distribution**

I. Introduction

Frank Knight makes distinct difference between the uncertainty and the disturbance [1]. The disturbance can be quantitatively estimated and the associated risk can be evaluated. In contrary, an uncertain observation cannot be quantified and is called to be "*pure and untainted*" uncertainty. Unknown effects exist, though they are frequently neglected or remain unnoticed. When their impact is significant, people start to do something. In case of unnoticeable events, no reaction is done. Once the distribution of a given disturbance is known, we may define and assess the associated risk. The distribution is unknown, and we may estimate its factors. Once the exact data distribution is not known, we only observe the effects. We may do some stochastic assumptions and proceed.

As data stochasticity plays (at least it should) an important role in the interpretation of observations classical approach to prediction, when we are only interested in the predicted expected (mean) value of a given variable may not be sufficient. Data variability may significantly falsify our predictions and the associated decisions will not be appropriate. Stochastic prediction has been introduced to address this certain issue [2]. The challenge of stochastic prediction is continuously evolving, driven by the necessity to have better understanding of uncertainty in complex systems. The paradigm of forecasting has undergone a shift from traditional deterministic point estimates to stochastic predictions. It is crucial in many real-world applications and it helps in decision-making under uncertainty. Control engineering is not an exception [3].

Stochastic prediction plays a crucial role in various fields, including finance, weather forecasting, and engineering, particularly in interference modeling in automation processes. The main challenge of stochastic prediction lies in accurately capturing the underlying uncertainty and variability of the system being modeled. Stochastic models incorporate probabilistic elements, allowing for a more realistic representation of uncertainty. This challenge is most acute in the tails of distributions. Most of the data used by machine learning (ML) models tends to cluster around the mean or median, extreme values are not sufficiently represented. This data-scarcity problem becomes a crisis in high risk environments, where sudden extreme events can cause significant damage.

In the case of interference modeling in automation processes, the typical approach relies on assumption that the parameters of distributions governing the interference are known and do not change in time - they are stationary. However, in practice, time series is non-stationary, and this nature of interference poses a significant challenge for accurate prediction, as models must adapt to these changes in real-time. The challenge is further compounded while dealing with heavy-tailed distributions [4], meaning extreme values occur more frequently than expected under light-tailed functions (e.g. normal) [5]. Ignoring these extreme observations from tails leads to significant underestimation of rare yet critical events.

Stochastic prediction, with its awareness to the tails of distributions, may provide valuable insights into rare but impactful events, such as extreme interference spikes that could disrupt automation processes. By accurately modeling the tails, stochastic prediction can help identify potential risks and enable proactive measures to mitigate their impact. Tail-aware stochastic prediction is crucial in domains, where one must accurately predict not just average behavior, but also the likelihood and severity of extreme observations.

The aim of this work is to explore and develop machine learning approaches for tail-aware stochastic prediction, with a specific focus on interference modeling in control engineering processes using LightGBMLSS (Light Gradient Boosting Machine for Location, Scale, and Shape) algorithm [6]. It extends Generalized Additive Models for Location Scale and Shape (GAMLSS) [7] by using gradient boosting. The XGBoostLSS framework [8] allows to utilize various distributions. Unfortunately, the most interesting distribution families [4]: α-stable or four parameter kappa (K4P) are not included due to their nonexistent or complex forms.

*This work was not supported by any organization

Institute of Control and Computational Engineering, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland : jan.szymczak2.stud@pw.edu.pl; pawel.domanski@pw.edu.pl

This work explores the accuracy of LightGBMLSS algorithm in modeling the tails of already implemented distributions [9], expands its capabilities to new distributions, evaluates the performance of the developed models using real-world datasets, and provides recommendations for applications.

This paper start with Section II describing the concept, followed by Section III, which describes XGBoostLSS and its extensions. The industrial data validation is described in Section IV, while Section V concludes the work.

II. Theoretical background

This research proposes the extension of the LightGBMLSS framework with two families of distributions: α-stable and four parameter Kappa. Traditionally, regression approach assumes process model, like ARMA (auto regressive moving average), or ARFIMA (auto regressive fractional-order integrated moving average) with added a random process (auxiliary variable) at the model output. An overwhelming majority of applications assumes stationary Gaussian noise, i.e. thin-tailed, as it may be easily identified [10].

Stochastic prediction allows to determine varying time series moments or quantiles. Therefore it incorporates non-stationarity as its fundamental assumptions.

A. Stationarity

A given time series X_i is considered stationary in a weak sense, if its mean, variance and auto-correlation remain constant in time, i.e. are time-independent.

The series X_i is called strictly stationary if joint distribution $(x_{i_1}, x_{i_2}, \ldots, x_{i_k})$ for each h and any finite sequence of $(i_1, i_2, \ldots, i_k)$ is identical to the $(x_{i_1+h}, x_{i_2+h}, \ldots, x_{i_k+h})$. Strict stationarity says that the joint distribution of a multidimensional variable is constant due to shifts in time.

Incremental stationarity means that the original time series is not stationary, buts its increments become stationary $x_i \sim \mathbb{I}(n)$. Trend stationarity says that the time series is treated as a sum of a deterministic trend and some stationary stochastic process, like white noise, $x_i = \alpha + \beta i + \epsilon_i$. Any process is said to be stationary around trend (or trend stationary) if it can be built as a linear combination of a stationary process and a trend. Therefore, after detrending, it starts to be stationary. Stationarity may be assessed with dedicated tests [4].

B. The α-stable family of distributions

The α-stable distributions [11] seem to be a natural selection for the heavy-tailed function. It covers various tail heaviness with normal probabilistic density function (PDF) being its special case. The distribution is defined as a closed PDF, an we use characteristics equation

$$\mathcal{F}^{\text{stab}}_{\alpha,\beta,\delta,\gamma}(x) = \exp\left\{ i\delta x - |\gamma x|^{\alpha} \left(1 + i\beta \frac{x}{|x|} l(x,\alpha)\right)\right\}, \qquad (1)$$

$$l(x,\alpha) = \begin{cases} \tan\left(\frac{\pi\alpha}{2}\right) & \text{for } \alpha \neq 1 \\ -\frac{2}{\pi}\ln|x| & \text{for } \alpha = 1 \end{cases},$$

with $\delta \in \mathbb{R}$ being a shift, $\gamma \geq 0$ a scale, $|\beta| \leq 1$ a skewness, and $0 < \alpha \leq 2$ a stability exponent. The function has four degrees of freedom: shift, scale, and two shape coefficients α and β. The shape factor α reflects tail heaviness. The smaller α is, the heavier tail is. For $\alpha < 2$ the second and higher moments do not exist. We get the following special cases:

- $\alpha = 2$ delivers independent realizations, specifically for $\alpha = 2;\ \beta = 0$ an exact normal distribution $\mathrm{N}\left(\delta, \sqrt{2}\gamma\right)$,
- $\alpha = 1;\ \beta = 0$ describes Cauchy PDF,
- $\alpha = 0.5;\ \beta = 1$ — Lévy,
- $\alpha = 1;\ \beta = 1$ — Landau,
- $\alpha = {}^{3}/_{2};\ \beta = 0$ — Holtsmark.

We may estimate its factors in many ways [12], it might challenging [13].

C. The four-parameter Kappa family of distributions

The K4F distribution [14] is given by

$$\mathcal{F}^{\text{K4P}}_{\xi,\beta,k,h}(x) = \frac{1}{\beta}\left[1 - \frac{k}{\beta}(x-\xi)\right]^{1/k-1} \left\{1 - h\left[1 - \frac{k}{\beta}(x-\xi)\right]^{1/k}\right\}^{1/h-1}, \qquad (2)$$

where $\xi \in \mathbb{R}$ is a shift, $\beta > 0$ a scale, while $h > 0$ and $k > 0$ are shape factors. The function exists in certain ranges:

$$\text{RANGE} = \begin{cases} -\infty < x \leq \xi + \frac{\beta}{k}, & \text{for } k > 0, \\ \xi + \beta\frac{1-h^{-k}}{k} < x < \infty, & \text{for } h > 0, \\ \xi + \frac{\beta}{k} \leq x < \infty, & \text{for } k < 0,\ h \leq 0. \end{cases} \qquad (3)$$

When $k \neq 0$, K4P reduces to: general Pareto $h = 1$, generalized extreme value $h = 0$ and generalized logistic for $h = -1.0$. When $k = 0$, it reduces to exponential with $h = 1$, to Gumbel with $h = 0$, and to logistic if $h = -1$. For $k = 1$, it reduces to uniform for $h = 1$ and to reverse exponential for $h = 0$. Four degrees of freedom allow to utilize K4P in many applications. we may estimate its factors using L-moments [15] and or regression [16].

III. Stochastic modeling and prediction

There are various machine learning approaches to model or predict distributions with tail-awareness. The first models used gradient boosting GAMLSS (Gradient Boosting for Additive Models for Location, Scale and Shape). Mayr *et al.* (2012) introduced a boosting algorithm for GAMLSS models, and Schlosser *et al.* (2019) later extended this to distributional regression forests [17]. XGBoost (eXtreme Gradient Boosting) was used to probabilistic forecasting (XGBoostLSS) in 2019, while in 2020 similar wors was done for CatBoost (Categorical Boosting) as CatBoostLSS [6].

These methods use boosting to estimate all parameters of a given distribution. The unified framework is summarized as Distributional Gradient Boosting Machines (e.g. LightGBMLSS). The likelihood-based boosting approach allows to predict entire conditional distributions, including tails. Moreover, boosting-based distributional models allow to predict one-sided and two-sided data. In Natural Gradient Boosting (NGBoost), the output for each prediction is a full probability distribution. NGBoost optimizes gradient of the likelihood function, allowing effective uncertainty capture.

The main advantage of NGBoost is its flexibility in choosing any parametric distribution for target. It was the first widely used boosting library and was a precursor to the later LightGBM/XGBoost LSS approaches [6].

Probabilistic Deep Learning has bee also adapted for probabilistic and tail-aware forecasting. There are approaches like DeepAR [17], which is an autoregressive recurrent network that predicts entire probability distributions for time series, however, in some cases, it has to assume a specific distributional form (e.g. Student-t) to handle heavy tails. Recent approaches use deep distributional regression models with extreme-value calibration to better capture tail behavior. For even better results these models combine neural networks with EVT (Extreme Value Theory) for tail extrapolation [18].

Ensemble and Bayesian methods [19] include Bayesian neural networks, Monte Carlo dropout or bootstrapped ensembles. These methods tend to focus more on general uncertainty quantification rather than explicit tail modeling. While they can capture some aspects of uncertainty, they may still assume Gaussian-like tails and residuals unless combined with heavy-tailed likelihoods. The downside of such methods is often high computational cost and complexity, as they need multiple model evaluations, whereas boosting methods provide direct distributional estimates in a single shot.

Each methodology illustrates the broad trend: accounting for uncertainty and tail risk has become a key focus in modern machine learning for time series and process modeling. However, there is still room for improvement in accurately capturing extreme events, especially in dynamic environments and while dealing with highly dispersed, intermittent demand data with many zeros and occasional large spikes.

Despite the fact that XGBoostLSS covers various PDFs its implementation does not support α-stable or four-parameter Kappa. Expanding XGBoostLSS with these functions is the key contribution of this research.

A. Implementation of α-stable distribution

The integration of any distribution into the LightGBMLSS framework requires the definition of a custom objective function that computes the gradients and Hessians of the NLL with respect to the distribution parameters. For standard distributions with closed-form PDFs, like Gaussian or Laplace, this is straightforward and computationally inexpensive. However, the α-stable distribution lacks a closed-form PDF. The density function is defined only as the inverse Fourier transform of its characteristic function:

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-itx} \phi(t) dt \tag{4}$$

Direct integration or Fast Fourier Transform (FFT) are computationally intensive. Especially, in the context of GBDT training, where gradients and Hessians must be re-evaluated over large datasets across many boosting iterations, standard numerical methods become impractical.

Numerical differentiation of the PDF would introduce significant numerical instability; particularly in tails, gradients may vanish or explode, preventing the boosting algorithm from converging properly. Thus, a direct implementation of the α-stable distribution in LightGBMLSS is not feasible in practice due to computational cost and severe numerical instability, and a surrogate model approach is developed.

To solve the issues described above, a neural network surrogate model is proposed to learn a fast, differentiable approximation of the α-stable log-PDF function, while retaining analytical precision in critical regions. The model first routes inputs based on their parameters:

- **Analytic Cauchy case:** for parameter configuration in a small, $\varepsilon = 10^{-3}$ neighborhood of the Cauchy case ($\alpha = 1$, $\beta = 0$), the model uses the exact, analytic log-PDF of the Cauchy distribution that is fully differentiable in PyTorch. It's done to achieve perfect accuracy for this numerically sensitive region that any neural-network surrogate model could not handle.
- **NN surrogate case:** for all other parameter combinations, the model uses a MLP with residual connections.

Neural network is trained to predict the log-PDF value $log(F^{stab}(x|\alpha, \beta, \delta, \gamma))$ given five inputs $(x, \alpha, \beta, \delta, \gamma)$. To improve learning efficiency, feature engineering is applied and model computes a six-dimensional feature vector $\big(z, \alpha, \beta, \text{sign}(z), \log(1 + |z|), z\beta\big)$, where $z = (x - \delta)/\gamma$.

This embeds the location-scale invariance of the distribution directly into the model, improving learning efficiency. Operating on raw inputs resulted in instability, especially in tails. To address this challenge, the model was trained using a curriculum learning strategy. A comprehensive dataset is generated in three bins by sampling thousands of points. Training follows a three-phase curriculum using these bins, starting from Bin 'A' containing light-tailed, near-Gaussian samples and progressively introducing more complex, skewed, and heavy-tailed points in Bins 'B' and 'C'. To further improve accuracy in tails, while maintaining it also near distribution's peak, composite Huber loss is used to explicitly up-weight these regions with fine-tuning at the very end of training.

It is evaluated on a large, independent test set to validate its accuracy with results shown in Table I. Three indexes are calculated: recursive mean square error (RMSE), mean absolute error (MAE) and median absolute error (MdAE) The model demonstrates excellent performance, with a median absolute error of 0.1103 and a global mean absolute error of 0.2980. The tails are obviously the most problematic region, but the model achieved promising MdAE of 0.1761.

TABLE I: Surrogate model log-PDF error metrics vs SciPy ground truth, for ≈200,000 points stratified by region.

Region	**N**	**MAE**	**RMSE**	**MdAE**
Global	199,837	0.2980	0.7513	0.1103
Center ($\lvert z\rvert \leq 2$)	79,574	0.2906	0.6875	0.0760
Shoulder ($2 < \lvert z\rvert \leq 5$)	59,945	0.2245	0.6216	0.1066
Tail ($\lvert z\rvert > 5$)	60,318	0.3808	0.9288	0.1761

As expected in case of α-stable family, the error distribution is heavy-tailed with some small number of extreme, numerically-challenging outliers. However, the overall surrogate's performance is not affected by them. Fig. 1 shows the

surrogate's output against the true log-PDF. Trained network acts as a fast and differentiable surrogate for the α-stable log-PDF, supporting gradient and Hessians computation.

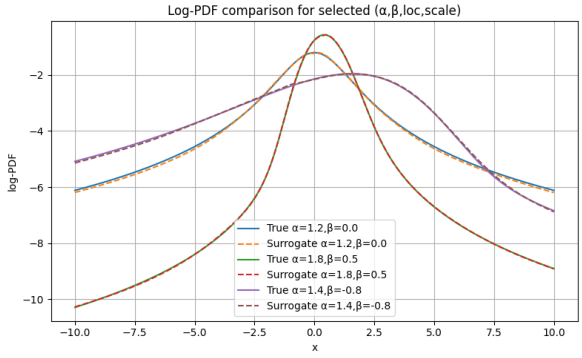


Fig. 1: Comparison of the surrogate model log-PDF (dashed line) against the true (solid line).

B. Implementation of K4P distribution

The main difference between integrating the K4P distribution and the α-stable distribution into LightGBMLSS is the availability of a closed-form PDF for the K4P family. Thus, no surrogate model is needed to ensure differentiability. Nevertheless, K4P remains challenging, because it depends on the parameter configuration through the constraints $w(x) = 1-\frac{k}{\beta}(x-\xi) > 0$ and $1-hw(x)^{1/k} > 0$. Moreover, the family includes several important limiting cases as $k \to 0$ and/or $h \to 0$. Numerically robust implementation must handle these limits and remain compatible with the LightGBMLSS.

The location is modeled as $\xi \in \mathbb{R}$, while the scale is constrained to $\beta > 0$ using a softplus transformation with a small offset. Two shape factors k and h are defined on $\mathbb{R}$. However, unconstrained estimation can lead to numerical instability. To address this issue, the implementation restricts the parameter space to $k \in [-1, 1]$ and $h \in [-1, 1]$ through hyperbolic tangent transformations. This interval covers all practical cases, excluding extremes that may lead to invalid results and unstable gradients. It preserves the functional form and interpretability of the K4P likelihood while reducing the probability of invalid parameter combinations.

Limiting regimes of the distribution stand as second source of numerical instability. The K4P family contains well-defined limits as $k \to 0$ and/or $h \to 0$, but the general expressions involve divisions by k and h and can become numerically unstable near these limits. To handle this, the PyTorch implementation switches to explicit limit formulas in small neighborhoods around zero:

- $k \to 0$ (with $h \neq 0$): the term $w(x)^{1/k}$ is replaced by $\exp(-y)$, where $y = \frac{x-\xi}{\beta}$. This yields a numerically stable representation of the CDF and log-density contributions without division by a vanishing k.
- $h \to 0$ (with $k \neq 0$): the CDF term converges to $F(x) = \exp(-R)$ with $R = w(x)^{1/k}$, which corresponds to the GEV limit. The log-density can be written in a stable form as $\log f(x) = -\log\beta + \left(\frac{1}{k} - 1\right)\log w(x) - R$.
- $k \to 0$ and $h \to 0$: the distribution reduces to Gumbel PDF. The implementation uses the closed-form Gumbel formulation in this regime.

All computations clamp intermediate values to safe ranges to prevent numerical overflow or underflow during training. Invalid-domain configurations are penalized via a differentiable quadratic barrier added to the log-likelihood, encouraging the model to avoid these regions during optimization. This approach preserves gradient information near the boundary and reduces the frequency of degenerate training behavior compared to hard masking or constant penalties.

IV. Industrial Data Validation

Validation uses real data from an industrial pulverized coal anonymous boiler from one-week period. Measurements include process variables: coal flow (F_c) and furnace pressure (Δp) and flue-gas emissions: O_2, CO, SO_2, NO_2, and dust concentration. There are approximately 10,000 observations.

Most variables fluctuate around an operating point. In contrast, CO is strictly non-negative and exhibits heavy-tail with occasional extreme spikes. Thus, it is treated as a one-sided variable. The rest is considered to be two-sided.

Each variable is modeled as a univariate time series using autoregressive lag taken from its own past values. Probabilistic forecasts are produced using LightGBMLSS by estimating full predictive distribution. Gaussian and Laplace distributions serve as baselines. Their results are compared with α-stable and K4P to assess whether they may more effectively capture skewness, kurtosis, and extreme events.

Figs. 2, 3, and 4 present sample plots for three representative variables: CO content (one-sided), furnace pressure (approximately Gaussian), and dust content (two-sided with significant right tail). Each figure shows the time series (top), marginal histogram (bottom-left), and autocorrelation for lags 1–240 minutes (bottom-right).

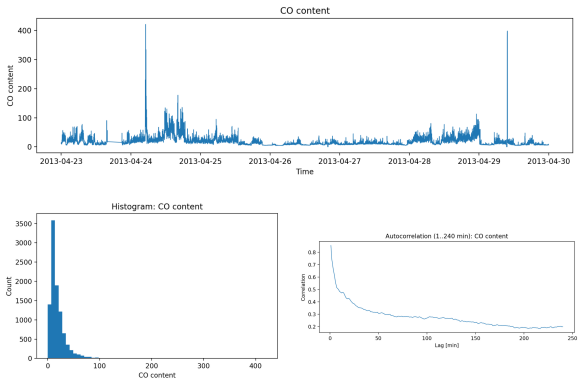


Fig. 2: CO content (one-sided)

These variables cover representative range of statistical properties, from near-Gaussian to strong skewness with heavy tails, making the data suited to assess different distributional assumptions. Table II summarizes results for two-sided variables. Besides basic metrics: Continuous Ranked Probability Score (CRPS) and Kolmogorov-Smirnov (KS),

two additional measures are evaluated: empirical coverage of nominal 50% and 90% (matching the nominal levels), and sharpness measured by an interquartile range (IQR).

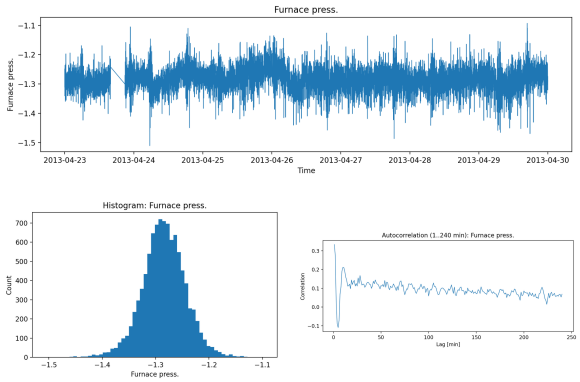


Fig. 3: Furnace pressure (two-sided)

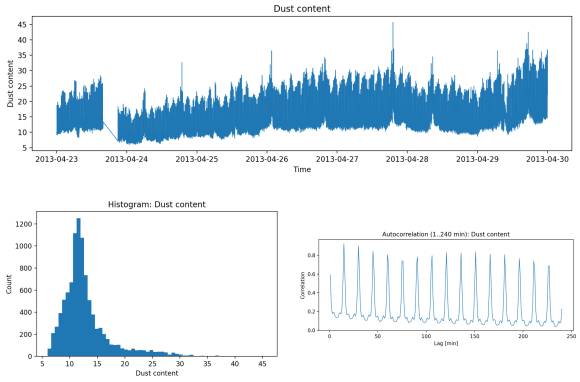


Fig. 4: Dust content (two-sided)

Sample good fit of α-stable distribution is shown for O_2, left 1 in Fig. 5. The predicted median closely tracks the observed values, capturing the main trends and fluctuations, even during sudden spikes or drops.

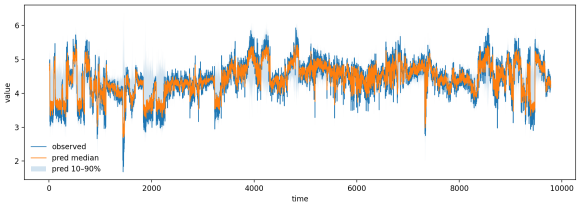
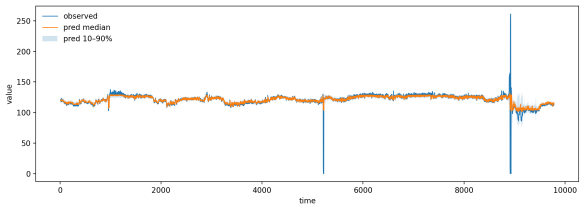


Fig. 5: Predicted vs observed O_2, left 1 – α-stable PDF

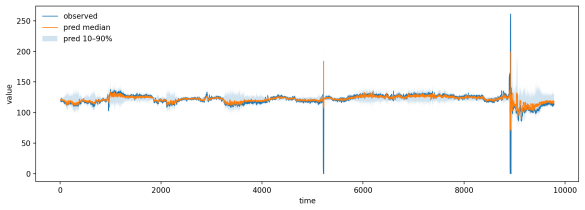
Fig. 6 compares the predictive medians and 10–90% predictive bands for SO_2 using Gaussian and α-stable PDFs. Gaussian model remains sharp through most of the series but substantially underestimates tails, with extreme spikes falling far outside the 10–90% band. In contrast, the α-stable model increases dispersion and shifts the predictive median upward around the largest spike, reflecting a heavier-tailed PDF that better accommodates rare extremes, but with reduced sharpness and higher CRPS.

TABLE II: Summary of probabilistic forecasting performance (mean $\pm$ std across folds). Best CRPSs are bold.

Target	PDF	CRPS	KS	Cov_{50}	Cov_{90}	IQR
F_c	Gaussian	**0.040 ± 0.009**	0.184 ± 0.112	0.570	0.936	0.108
	Laplace	0.042 ± 0.003	0.188 ± 0.026	0.778	0.992	0.150
	α-stable	0.046 ± 0.006	0.123 ± 0.131	0.611	0.950	0.540
	K4P	0.062 ± 0.017	0.190 ± 0.152	0.469	0.808	0.126
O_2, L1	Gaussian	**0.134 ± 0.002**	0.119 ± 0.068	0.540	0.910	0.331
	Laplace	0.156 ± 0.004	0.190 ± 0.027	0.827	0.997	0.598
	α-stable	0.148 ± 0.012	0.105 ± 0.074	0.554	0.933	0.400
	K4P	0.240 ± 0.053	0.200 ± 0.134	0.518	0.822	0.527
O_2, L2	Gaussian	**0.128 ± 0.003**	0.069 ± 0.040	0.529	0.893	0.300
	Laplace	0.160 ± 0.006	0.199 ± 0.031	0.844	0.998	0.635
	α-stable	0.155 ± 0.015	0.146 ± 0.045	0.590	0.924	0.309
	K4P	0.270 ± 0.044	0.140 ± 0.098	0.499	0.876	0.662
O_2, R1	Gaussian	**0.106 ± 0.003**	0.052 ± 0.033	0.534	0.903	0.255
	Laplace	0.142 ± 0.011	0.207 ± 0.033	0.882	0.998	0.590
	α-stable	0.153 ± 0.018	0.135 ± 0.076	0.729	0.982	0.554
	K4P	0.236 ± 0.039	0.223 ± 0.127	0.503	0.856	0.520
O_2, R2	Gaussian	**0.127 ± 0.005**	0.051 ± 0.011	0.524	0.896	0.299
	Laplace	0.157 ± 0.005	0.204 ± 0.030	0.869	0.997	0.639
	α-stable	0.150 ± 0.008	0.140 ± 0.034	0.646	0.973	0.471
	K4P	0.238 ± 0.015	0.172 ± 0.073	0.497	0.859	0.521
dust	Gaussian	1.656 ± 0.123	0.174 ± 0.036	0.585	0.865	2.969
	Laplace	**1.372 ± 0.075**	0.123 ± 0.029	0.461	0.855	1.691
	α-stable	1.479 ± 0.051	0.204 ± 0.018	0.519	0.875	2.022
	K4P	1.851 ± 0.248	0.196 ± 0.046	0.372	0.740	2.706
Δp	Gaussian	**0.022 ± 0.001**	0.052 ± 0.026	0.509	0.893	0.051
	Laplace	0.022 ± 0.001	0.112 ± 0.069	0.434	0.926	0.043
	α-stable	0.023 ± 0.003	0.320 ± 0.144	0.205	0.409	0.024
	K4P	0.025 ± 0.001	0.184 ± 0.088	0.372	0.730	0.042
SO_2	Gaussian	**2.025 ± 1.347**	0.167 ± 0.056	0.420	0.743	3.644
	Laplace	2.396 ± 2.295	0.160 ± 0.084	0.413	0.840	7.202
	α-stable	2.918 ± 1.479	0.193 ± 0.086	0.603	0.978	5.847
	K4P	4.605 ± 2.740	0.282 ± 0.034	0.478	0.792	6.866
NO_2	Gaussian	2.411 ± 0.443	0.058 ± 0.046	0.479	0.857	4.148
	Laplace	**2.356 ± 0.339**	0.112 ± 0.031	0.365	0.826	2.917
	α-stable	4.111 ± 0.793	0.143 ± 0.025	0.690	0.977	11.474
	K4P	4.768 ± 1.295	0.162 ± 0.120	0.463	0.858	10.813



(a) Gaussian



(b) α-stable

Fig. 6: Predictive medians and 10–90% bands for SO_2

The CO is strictly non-negative and exhibits strong right-skewness with occasional extreme spikes (see Fig. 2). The CO is modeled in a transformed space using $y_{\text{model}} = \log(1+y_{\text{raw}})$, while forecasts are presented in the original scale by inverting transformation. Table III shows CRPS both in raw and log-transformed spaces, where extreme observations are compressed and average accuracy is addressed.

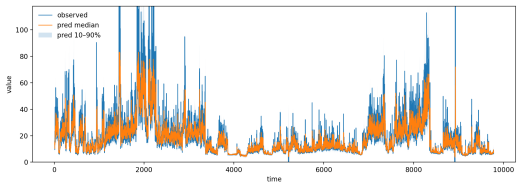
Gaussian model achieves the lowest CRPS with the best average accuracy. However, its empirical coverage is below nominal, suggesting underdispersion under heavy tails.

TABLE III: Forecasting performance for CO (mean $\pm$ std across folds). Best (lowest) CRPS per column is bold.

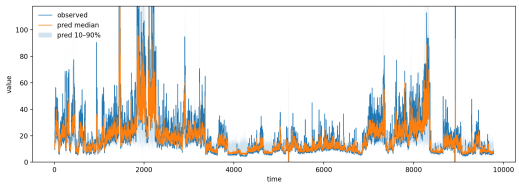
PDF	CRPS_{raw}	CRPS_{log}	KS	Cov_{50} / Cov_{90}	IQR
Gaussian	**2.814 $\pm$ 1.003**	**0.141 $\pm$ 0.013**	0.144 $\pm$ 0.083	0.488 / 0.895	5.542
Laplace	3.443 $\pm$ 0.890	0.183 $\pm$ 0.009	0.221 $\pm$ 0.025	0.822 / 0.995	11.385
α-stable	3.154 $\pm$ 1.025	0.160 $\pm$ 0.023	0.162 $\pm$ 0.072	0.542 / 0.957	6.977
K4P	6.927 $\pm$ 3.471	0.385 $\pm$ 0.209	0.367 $\pm$ 0.115	0.304 / 0.766	13.071

Similarly to previous variables, α-stable PDF increases tail flexibility and improves the coverage, but at a cost due to the trade-off between tail-risk representation and averaged accuracy. In contrast, Laplace produces the widest predictive intervals and the highest coverage, but with no improvement in CRPS. K4P again underperforms across all metrics.

Fig. 7 visualizes prediction bands. Gaussian model is quite sharp, but fails to fully capture the magnitude and frequency of extreme CO spikes, whereas the α-stable model produces heavier-tailed forecasts that better cover spike episodes.



(a) Gaussian



(b) α-stable

Fig. 7: Predictive medians and 10–90% predictive bands for CO content (one-sided, ppm scale).

V. Conclusion and Further Research

The extension of LightGBMLSS with α-stable and K4P distributions proved to be a valuable. Validation demonstrates benefits, particularly in scenarios involving heavy-tails and extreme events. However, some limitations are observed. The surrogate neural network model of α-stable PDF, while effective, introduces additional complexity and potential errors. Its accuracy is crucial. Knowing that, the model achieves satisfactory performance outperforming other PDFs in several prediction scenarios. Nonetheless its further refinement is necessary to ensure the reliability.

Incorporating more complex distributions, especially those requiring surrogate models, may introduce additional computational overhead. It is observed that training times increases with the complexity of the PDF, particularly for the α-stable function. Optimization of the hyper-parameters for such models requires more effort and time, which could limit its use. Future work should focus on optimizing the computational efficiency to ensure its scalability and usability.

The most significant limitation of this study lies in the implementation of the K4P. The current implementation exhibits poor performance, likely due to numerical instability and complexity. This highlights the need for further research of the K4P to improve its practical applicability. The use of truncated K4P might also help.

This study focuses on one-dimensional data. Extending the framework to handle multivariate distributions is an important future research.

References

[1] F. Knight, *Risk, Uncertainty, and Profit*. Boston MA: Hart, Schaffner and Marx, Houghton Mifflin, 1921.

[2] S. Haben, M. Voss, and W. Holderbaum, "Probabilistic forecast methods," in *Core Concepts and Methods in Load Forecasting: With Applications in Distribution Networks*. Cham: Springer International Publishing, 2023, pp. 201–227.

[3] P. Domański, "Non-gaussian properties of the real industrial control error in SISO loops," in *Proceedings of the 19th International Conference on System Theory, Control and Computing*, 2015, pp. 877–882.

[4] P. D. Domański, *Back to Statistics. Tail-Aware Control Performance Assessment*. Boca Raton, FL: CRC Press, Taylor & Francis Group, 2025.

[5] N. Taleb, "Statistical consequences of fat tails: Real world preasymptotics, epistemology, and applications," 2022, arXiv:2001.10488v3 [stat.OT], Cornell University Library.

[6] A. März and T. Kneib, "Distributional gradient boosting machines," 2022, arXiv:2204.00778v1 [stat.ML]. [Online]. Available: https://arxiv.org/abs/2204.00778

[7] R. A. Rigby and D. M. Stasinopoulos, "Generalized additive models for location, scale and shape," *Journal of the Royal Statistical Society Series C: Applied Statistics*, vol. 54, no. 3, pp. 507–554, 2005.

[8] A. März, "LightGBMLSS: An Extension of LightGBM to Probabilistic Modelling," https://github.com/StatMixedML/LightGBMLSS, 2023, gitHub repository, Version 0.4.0.

[9] A. März, "XGBoostLSS – an extension of XGBoost to probabilistic forecasting," 2019, arXiv:1907.03178 [stat.ML]. [Online]. Available: https://arxiv.org/abs/1907.03178

[10] R. Isermann and M. Münchhof, *Identification of Dynamical Systems*. Berlin, Heidelberg: Springer-Verlag, 2009.

[11] C. Nikias and M. Shao, *Signal processing with alpha-stable distributions and applications*. New York, NY: Wiley-Interscience, 1995.

[12] J. Royuela-del-Val, F. Simmross-Wattenberg, and C. Alberola-Lopez, "libstable: fast, parallel, and high-precision computation of α-stable distributions in R, C/C++, and MATLAB," *Journal of Statistical Software, Articles*, vol. 78, no. 1, pp. 1–25, 2017.

[13] X. Liao and P. Domański, "On estimation of α-stable distribution using L-moments," *Fractal and Fractional*, vol. 9, no. 11, 2025.

[14] J. Hosking, "The four-parameter kappa distribution," *IBM Journal of Research and Development*, vol. 38, no. 3, pp. 251–258, 1994.

[15] J. Hosking and J. R. Wallis, *Regional Frequency Analysis: An Approach Based on L-Moments*. Cambridge, UK: Cambridge University Press, 1997.

[16] L. da Silva Costa and F. Ferraz do Nascimento, "Regression in extremes using the four-parameter kappa distribution," *Communications in Statistics - Simulation and Computation*, vol. 0, no. 0, pp. 1–13, 2023.

[17] G. Papacharalampous and H. Tyralis, "A review of machine learning concepts and methods for addressing challenges in probabilistic hydrological post-processing and forecasting," *Frontiers in Water*, vol. 4–2022, 2022.

[18] O. C. Pasche, "Neural networks for extreme quantile regression with an application to forecasting of flood risk," 2024, arXiv:2208.07590v4 [stat.ME].

[19] H. Chipman, E. George, and R. Mcculloch, "Bayesian ensemble learning," in *Advances in Neural Information Processing Systems*, B. Schölkopf, J. Platt, and T. Hoffman, Eds. MIT Press, 2006, vol. 19.

[20] M. Giełczewski, M. Piniewski, and P. Domański, "Mixed statistical and data mining analysis of river flow and catchment properties at regional scale," *Stochastic Environmental Research and Risk Assessment*, vol. 36, pp. 2861–2882, 2022.