A Mean Variance Estimation Model for Predicting Total Load Shedding and Assessing Uncertainty in Cascading Power System Failures

Antonis Georgiou[1], George Paphitis[2], Panagiotis N. Papadopoulos[3], Mathaios Panteli[2]
[1]Department of Mathematics, ETH Zurich, Switzerland
[2]KIOS Center of Excellence, Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus
[3]School of Electrical and Electronic Engineering, The University of Manchester, Manchester, United Kingdom
ageorgiou@student.ethz.ch
[pafitis.g.georgios, panteli.mathaios]@ucy.ac.cy
panagiotis.papadopoulos@manchester.ac.uk

***Abstract*—Cascading failures in power systems are observed with increasing frequency, and the complexity of modern grids is making real-time severity assessment challenging. Machine learning models have been employed to predict load shedding resulting from such events; the use of single-point predictions and evaluation based solely on average accuracy metrics is considered unreliable, as large errors may be obscured and predictive risk cannot be quantified. In this work, an ensemble of Mean Variance Estimation networks combined with conformal prediction is proposed to enhance predictive performance and provide calibrated uncertainty estimates. Through this framework, reliable prediction intervals (PIs) are generated to enable risk-aware decision-making for real-time operation and planning. It is demonstrated that the proposed model outperforms commonly used approaches such as deep neural networks and eXtreme Gradient Boosting, achieving a mean absolute error of 0.0469 and root mean squared error of 0.0939. It attains 95.3% coverage with an average PI width of 0.1943, 0.10 narrower than those of other models, indicating improved accuracy and greater reliability.**

***Index Terms*—Cascading Failures, Load Shedding, Machine Learning, Reliability, Resilience**

I. Introduction

The increasing penetration of inverter-based resources (IBRs), the rise of large data center loads, and the limited flexibility of grid infrastructure present significant operational challenges for power systems [1]. These conditions amplify variability, weaken system inertia, and increase the likelihood of cascading failures, introducing significant uncertainty in grid response under stressed operating conditions [1], [2]. This leads to two critical questions: to what extent can load shedding (LS), induced by cascading events under varying loading conditions and contingencies, be reliably predicted and quantified, and how reliable are the average metrics reported in the literature when evaluating model performance under such conditions. To address these challenges, this work proposes a mean–variance estimation model (MVE) to jointly predict total LS and assess predictive uncertainty during cascading power system failures under varying loading levels.

Most existing studies on cascade-induced LS rely on point-estimate machine learning (ML) models, meaning that they provide a single value for the expected LS [3], [4]. Furthermore, these models are evaluated using average metrics such as mean absolute error (MAE) and root mean squared error (RMSE). While these metrics indicate general performance trends, they offer limited insights into the reliability of individual predictions. For critical infrastructures like power systems, it is not sufficient to provide a single prediction or average metrics; operators must understand the full range of potential outcomes and the probability that the true value deviates from the predicted interval. Many studies employ Bayesian frameworks to quantify uncertainty in ML models [5], [6]. Dropout-based approaches have also been proposed to approximate Bayesian inference and extract predictive uncertainty [7], [8]. However, both Bayesian and dropout-based variational approaches inherently rely on prior assumptions. These methods can yield reliable uncertainty estimates when the chosen prior reflects the underlying system. In practical applications, however, the true prior is rarely known, and a misspecified prior may lead to poorly calibrated or misleading confidence levels [9]. Furthermore, ML models do not inherently produce calibrated prediction intervals (PIs). They are optimized for point accuracy and often exhibit overconfident predictions that do not reflect true uncertainty [9]–[11].

In response to these limitations, conformal prediction (CP) [12] has gained increasing attention [13] as a model-agnostic framework that provides statistically valid PIs without requiring prior assumptions. However, CP does not capture epistemic uncertainty in non-stationary systems such as power systems. To address this limitation, a deep ensemble of mean–variance estimation (MVE) networks [14] is combined with CP to

This work was supported by the Horizon Europe programme through the project "Solid Preparedness and Resilience for Robust Operations during Disaster Wilderness" (SPARROW, Grant Agreement No. 101168499) 979-8-3315-2503-3/25/$31.00 © 2025 IEEE.

predict total LS and provide input-adaptive, well-calibrated PIs. Deep ensembles of MVE networks show superior out-of-distribution detection by expressing higher uncertainty on such samples [14], which is crucial for anomaly detection.

This work employs an ensemble of MVE neural networks combined with CP to predict total LS resulting from cascading failures under varying system loading conditions and extreme contingencies ranging from $N-2$ to $N-16$. The contingencies along with the loading conditions are passed to the AC-Cascading Failure Model [15] to calculate the cascade-induced LS. In addition to prediction, the ensemble quantifies uncertainty, enabling assessment of model reliability and determining whether average performance metrics underestimate prediction error, which may lead to over-optimistic conclusions and increased operational risk.

The main contributions of this work are:

- Development of an ensemble of MVE neural networks for predicting total LS induced by cascading failures while simultaneously generating well-calibrated and adaptive PIs with theoretical guarantees.
- Evaluation of predictive uncertainty to show that commonly used average performance metrics may not accurately reflect model reliability under varying loading conditions and extreme contingencies.

The remainder of the paper is structured as follows. Section II presents the methodology developed for predicting total load shedding under uncertainty. Section III describes the network model, the data used, the adaptations applied to the methodology for this study, and the corresponding results. Finally, Section IV concludes the paper.

II. Methodology for Predicting Total Load Shedding Under Uncertainty

The first step of the proposed methodology (depicted in Fig. 1) involves generating the dataset required for training the ML models. The target variable, total LS, is computed using the AC-CFM, which has been validated following the procedures established by the IEEE PES Working Group on Cascading Failures. The input features are obtained by solving an AC-OPF at the cascade onset. Subsequently, multiple ML models are trained and evaluated using standard average performance metrics, and their prediction uncertainty is assessed.

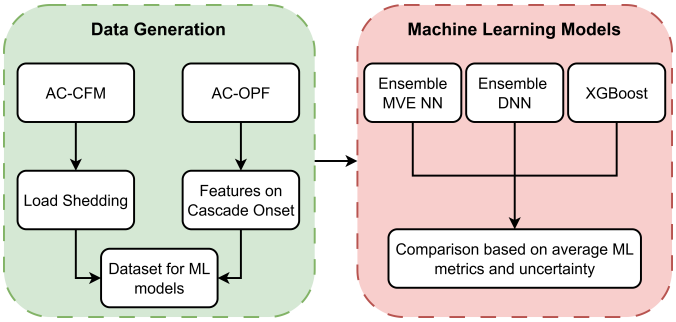

Fig. 1: Methodology for Data Generation and Model Comparison.

A. Data Generation

In this work, all $N-1$, $N-2$, and $N-3$ outages are generated. To include high-impact, low-probability (HILP) events, contingencies of order $\geq N-3$ are added using a stochastic weather-based fragility model [16]. The resulting contingency set is evaluated under multiple loading conditions using the AC-CFM to simulate cascading propagation.

The AC-CFM is a quasi–steady-state cascading failure model that includes common protection mechanisms such as under-frequency LS and over-frequency generator tripping, undervoltage LS, excitation limits, and line overloading protection. At each cascade step, it solves an AC power flow and applies the corresponding protection actions. As failures propagate, the network may split into islands; the model is applied recursively to each island until the cascade stops, and the total LS is then computed.

To prevent data leakage, all input features are extracted from the AC-OPF solution at the onset of the cascade, before any subsequent outages occur. The feature set includes line status, bus voltage magnitude and angle, active and reactive power injections, power flows, and loading levels. Cascade-induced LS is then obtained by evaluating each initiating contingency using the AC-CFM.

B. Machine Learning Models

A MVE network is a model that outputs two values: the prediction $\mu_\theta(x)$ and variance $\sigma_w^2(x)$. The data is modeled as

$$y_i = \mu_\theta(x_i) + \varepsilon_i, \tag{1}$$

where $x_i \in \mathbb{R}^d$, $d \geq 1$ are the features, ε_i the prediction error, and it is assumed that $\varepsilon_i \sim \mathcal{N}(0, \sigma_w^2(x_i))$, which makes the target values $y_i \sim \mathcal{N}(\mu_\theta(x_i), \sigma_w^2(x_i))$. We fit a MVE network on the data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, by maximizing the likelihood

$$\mathcal{L}(\theta, w; \mathcal{D}) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\ \sigma_w^2(x_i)}} \exp\left(\frac{y_i - \mu_\theta(x_i)}{\sigma_w^2(x_i)}\right), \tag{2}$$

or equivalently, minimizing the negative log-likelihood (NLL)

$$\begin{aligned}\mathcal{L}_{\text{NLL}}(\theta, w) &= -\log \mathcal{L}(\theta, w; \mathcal{D}) \\ &= \sum_{i=1}^{N} \left(\frac{(y_i - \mu_\theta(x_i))^2}{2\sigma_w^2(x_i)} + \frac{1}{2}\log\left(2\pi\sigma_w^2(x_i)\right)\right),\end{aligned} \tag{3}$$

with respect to the network parameters (θ, w). We train the MVE network as suggested in [17]. The model consists of two independent sub-networks; one for predicting the mean $\mu_\theta(x)$, and one for predicting the variance $\sigma_w^2(x)$. As explored in [17], a warm-up period of only updating the parameters of the mean, and keeping the parameters of the variance sub-network constant is used. This generally helps with convergence, as stated in [17]. Intuitively, this instability could be caused by the model achieving good NLL scores just from updating the variance at early stages of training where the parameters θ are still in regions close to their random initialization. After the

network has achieved a good enough estimate of the mean, the variance parameters begin to be updated, either together with the mean or by holding the parameters of the estimated mean fixed. Finally, for the complete model, $M \geq 1$ MVE networks are trained as described above, and their predictions are combined following the approach in [14],

$$\mu_\star(x) = \frac{1}{M}\sum_{m=1}^{M}\mu_{\theta_m}(x), \tag{4}$$

and

$$\sigma_\star^2(x) = \frac{1}{M}\sum_{m=1}^{M}\left(\sigma_{w_m}^2(x) + \mu_{\theta_m}^2(x)\right) + \mu_\star^2(x), \tag{5}$$

where $(\theta_m, w_m),\ m = 1,\ldots,M$, denote the parameters of the trained $m-th$ MVE network. Thus, the final output is $\left(\mu_\star(x), \sigma_\star^2(x)\right)$. For a fixed input x_i, suppose each model makes some error $\varepsilon_m = y_i - \mu_{\theta_m}(x_i)$, and thus $\varepsilon_m \sim \mathcal{N}\left(0, \sigma_{\theta_m}^2(x_i)\right)$, $m = 1,\ldots,M$. The model-averaging ensemble impacts the expected squared error as follows

$$E(y_i - \mu_\star(x_i))^2 = \text{Var}(y_i - \mu_\star(x_i)) \tag{6}$$

$$= \text{Var}\left(\frac{1}{M}\sum_{m=1}^{M}\varepsilon_m\right)$$

$$= \frac{1}{M^2}\left(\sum_{m=1}^{M}\text{Var}(\varepsilon_m) + \sum_{m\neq m'}\text{Cov}(\varepsilon_m, \varepsilon_{m'})\right) \tag{7}$$

We can lower the expected square error by enforcing the models in the ensemble to have less correlated errors. For example, random initialization of the parameters θ or using relatively high dropout rates may help reduce the correlation of errors. Negatively correlated errors hardly occur in practice, since the models still have the same architecture and are trained on the same data, which makes it unlikely that a model under-predicts where another over-predicts.

In addition to the proposed MVE network, two widely used baseline models, Deep Neural Networks (DNN) and eXtreme Gradient Boosting (XGBoost) [18], were implemented. These models are included solely for comparison to highlight the added value of the proposed MVE framework.

C. Conformal Prediction

Conformal Prediction (CP) is a simple yet powerful method of assessing the uncertainty of a model's predictions in distribution-free settings. Suppose that the input–target pairs (x_i, y_i) are independent and identically distributed (i.i.d.) according to a distribution P. The idea is that, given a trained predictor $\hat{\mu}_\theta(\cdot)$ and an input $x_i \in \mathcal{X}$, a confidence region $\hat{C}(x_i)$ is to be constructed such that

$$P(y_i \in \hat{C}(x_i)) \geq 1 - \alpha, \tag{8}$$

for a given $\alpha \in (0, 1)$.

To accomplish this, first let D_{train}, D_{cal}, D_{test} be a partition of the data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, and train any predictor $\mu_\theta(\cdot)$ on D_{train}. Then, define $r_i = |\mu_\theta(x_i) - y_i|$, for $(x_i, y_i) \in D_{cal}$ to be the absolute residuals of the trained model on the *calibration set*. Since r_i are $i.i.d$, their ranks are uniformly distributed. In other words, the probability that some $r_i \in D_{cal}$ is the kth smallest among all r_i's in D_{cal} is $1/n_{cal}$, where $n_{cal} := |D_{cal}|$. If we define

$$\hat{q} := \text{the } \lceil (1-\alpha)(n_{cal}+1) \rceil \text{ smallest of } r_i \in D_{cal}, \tag{9}$$

then their ranks being uniformly distributed, (and assuming that there are almost surely no ties between r_i's) implies that

$$P(r \leq \hat{q}) = \lceil (1-\alpha)(n_{cal}+1) \rceil / (n_{cal}+1) \tag{10}$$

which achieves the goal of

$$P\left(y \in [\mu_\theta(x) \pm \hat{q}]\right) \in \left[1-\alpha, 1-\alpha+\frac{1}{n_{cal}+1}\right), \tag{11}$$

for some $r = |y - \mu_\theta(x)|, (x, y) \in D_{test}$.

In general, $r_i = s(x_i, y_i)$ may be defined, where $s(\cdot,\cdot)$ is a function that assigns lower scores to better predictions and higher scores to worse predictions. This method of using an additional out-of-sample set D_{cal} is called split CP and it generally works under the weaker assumption of exchangeability. The reason a calibration set is needed to learn $\hat{q}$ (or a set function $\hat{C}(\cdot)$) instead of simply learning it on the training set, is that any out of sample residual calculated on a point of D_{test}, will no longer be independent with the residuals of the training set. For further details and reading on CP the interested reader is referred to [12].

III. Case Study Application

The test network used in this study is the IEEE 39-bus system. This network consists of 39 buses, 46 branches and 10 generators. In total, there are 16,679 scenarios in the dataset (46 N-1, 1,035 N-2, 15,180 N-3, and 582 HILP events ranging from $N-4$ to $N-16$). These scenarios are evaluated for 5 load levels, 60%, 70%, 80%, 90% and 100% of the peak load. Therefore, the dataset consists of 83,395 contingency scenarios. After filtering the dataset to include only cases where the AC-OPF converged successfully, it consists of 80,569 contingency scenarios.

The feature set includes 329 variables: 46 binary contingency indicators (1 = in service, 0 = out of service), one system loading value, 184 branch flow features (active/reactive power at both ends of 46 lines), 20 generator injection values (10 active and 10 reactive), and 78 bus voltage features (39 magnitudes and 39 angles). Therefore, for equation (1), the feature vector x_i is 329-dimensional ($x_i \in \mathbb{R}^{329}$) and normalized targets $y_i \in [0, 1]$, for $i = 1, 2, \ldots, N$, where $N = 80,569$ is the number of contingency scenarios.

Through experimentation, the best-performing model was identified as an ensemble of $M = 5$ MVE networks, each consisting of two independent sub-networks with 7 layers and 512 hidden units. After each layer, a ReLU activation function, Batch-Normalization layer, and dropout were used. The dropout rate was set to 0.3 for the mean sub-network and 0.1 for the variance sub-network, as this not only prevents overfitting but also enhances the predictive power of the

TABLE I: Performance metrics for ensemble models and baseline methods

Model	R^2	RMSE	MAE	Coverage	Width
5 MVE Ens.	0.8288	0.0939	0.0469	95.3%	0.1943
5 DNN Ens.	0.8288	0.0939	0.0469	95.0%	0.3074
1 MVE Ens.	0.8175	0.0970	0.0544	95.3%	0.2186
1 DNN Ens.	0.8175	0.0970	0.0544	95.2%	0.3259
XGBoost	0.7210	0.1198	0.0787	94.6%	0.6899

ensemble by reducing error correlation across models (see Equation (5)). For the output of the variance sub-network (variance predictor), the exponential function $\phi(x) = e^x$ is used, as it ensures positive outputs and provides improved training stability. In addition, the NLL loss function (3) was modified to penalize exploding variances by adding the term

$$\mathcal{L}_{\text{NLL}-\beta}(\theta, w) = \mathcal{L}_{NLL}(\theta, w) + \sum_{i=1}^{N} \frac{\beta}{\sigma_w^2(x_i)}, \qquad (12)$$

and $\beta = 10^{-4}$ was used. During training, a two-phase strategy was employed to optimize the sub-networks predicting both the mean $\mu_\theta(x_i)$ and variance $\sigma_w^2(x_i)$. In the initial phase, variance outputs were fixed at $\sigma_w^2(x_i) = 1$ while only the mean predictor parameters were updated. After 80 epochs of mean-only training, a transition was made to a second phase in which the variance parameters were optimized while the mean predictor was kept frozen. This approach led to superior performance compared to joint optimization from initialization. Both sub-networks were trained using the AdamW optimizer. We applied ℓ_2 regularization with weight decay of 10^{-3} across all parameters. The learning rates were set to 10^{-3} for the mean predictor and 10^{-5} for the variance predictor, reflecting different sensitivities of the components during optimization.

The predictive performance of the proposed model is summarized in Table I (5 MVE Ens.) . In domains involving critical infrastructure, such as the power grid, it is essential not only to achieve high predictive accuracy but also to quantify the model's confidence in its predictions. Reliable uncertainty estimates are crucial for informed decision-making.

The Gaussian assumption in Equation (1) is not satisfied in practice [1]. As a result, using a Gaussian quantile $z_{\alpha/2}$ to construct confidence intervals of the form $y_i \in \left[\mu_\star(x_i) \pm z_\alpha \sqrt{\sigma_\star^2(x_i)}\right]$ is not statistically justified. However, the NLL remains a meaningful loss function. When the predicted variance is kept constant, minimizing the NLL is equivalent to minimizing the standard mean squared error (MSE). Furthermore, if the mean is fixed, optimizing the NLL with respect to the variance encourages the model to assign higher uncertainty to predictions where the mean differs significantly from the target. This property makes the NLL a useful objective even when the true underlying distribution deviates from the Gaussian.

[1]In our case, the target values y_i are clearly non-Gaussian since negative LS values are not possible

To produce valid $(1-\alpha)$ PIs for the targets y_i, CP was used. On a held-out calibration set D_{cal}, the normalized residuals of the model are computed; that is, using the function

$$r_i = s(x_i, y_i) = \frac{|y_i - \mu_\star(x_i)|}{\sqrt{\sigma_\star^2(x_i)}}, \qquad (13)$$

the $\lceil (1-\alpha)(n_{cal}+1) \rceil$ smallest value of the set

$$\{s(x_i, y_i) : (x_i, y_i) \in D_{cal}\}$$

is selected and denoted as q_a. Then, PIs for the test set are generated as $[\mu_\star(x_i) \pm q_\alpha \sqrt{\sigma_\star^2(x_i)}]$. An important advantage of normalizing the residuals using a variance predictor is that adaptive PIs are obtained. For each input $x \in \mathbb{R}^d$, a PIs of potentially different width is obtained, enabling the confidence of the model in its own prediction to be inferred. On the other hand, when absolute residuals are used as in Section II-C, a constant-width prediction band is obtained across all predictions. In practice, the method is expected to achieve at least the desired $(1-\alpha)$ coverage while producing PIs that are as narrow as possible.

Table I compares different models in terms of their predictive capabilities and uncertainty quantification. "M MVE Ens." is the ensemble of MVE networks described in the previous section, and "M DNN Ens." is an ensemble of only the mean subnetworks (no variance predictor). In cases where no variance predictor was available the absolute residuals for CP (constant width PIs) were used. The "Width" column corresponds to the average interval length. The takeaway from Table I is that using a variance predictor trained via NLL minimization yields higher empirical coverage with substantially tighter PIs (95.3% coverage with an interval width of 0.2186, compared to 95.2% and 0.3259 for the DNN, and 94.6% and 0.6899 for XGBoost). This indicates higher confidence in the predictions of the proposed model.

Furthermore, the use of an ensemble, rather than a single model, not only improves predictive accuracy but also leads to narrower intervals on average. Specifically, the 5-member MVE ensemble achieved identical point prediction performance to the 5-member DNN ensemble in terms of R^2 (0.8288), RMSE (0.0939), and MAE (0.0469), while providing superior uncertainty quantification with 95.3% coverage and an interval width of 0.1943, compared to 95.0% coverage and a width of 0.3074 for the DNN ensemble. Additionally, the 5-member MVE ensemble outperformed both single-model MVE and DNN ensembles (R^2 = 0.8175, RMSE = 0.0970, MAE = 0.0544) as well as XGBoost (R^2 = 0.7210, RMSE = 0.1198, MAE = 0.0787), demonstrating the clear benefits of ensembling in both predictive accuracy and uncertainty reliability. For all models, the PIs were clipped to be in [0,1], since LS values cannot be negative or exceed 100%.

Figure 2 presents the calibrated CP intervals of the best-performing model (5 MVE Ensemble) across different load levels. The model maintains well-calibrated intervals in all cases, with the majority of true values lying within the predicted ranges. Each cluster corresponds to a distinct load level, where predicted means are shown in color, actual values

TABLE II: Average Confidence Interval Widths by Load Level for the 5 MVE Ens. Model

Load Level	Average Interval Width
1.0	0.3201
0.9	0.2425
0.8	0.1661
0.7	0.1301
0.6	0.1181

in black, and shaded regions denote the 95% PIs. As reported in Table II, the average interval width increases with load level, reflecting greater uncertainty under stressed system conditions. This behavior arises because higher loading leads to more variable LS outcomes, while lower loading results in more predictable responses. These findings highlight the necessity of adaptive PIs rather than fixed-width bands.

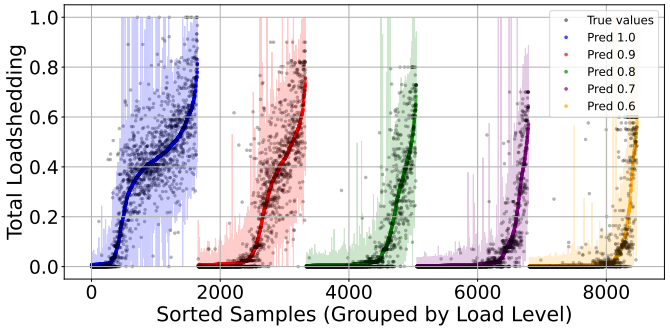

Fig. 2: Conformal PIs for total LS across all load levels, using the proposed model 5 MVE Ens.

IV. Conclusion

This work employed an MVE network combined with CP to predict cascade-induced LS and quantify predictive uncertainty. The MVE model was benchmarked against DNN and XGBoost using accuracy metrics (R^2, RMSE, MAE) and uncertainty metrics (coverage, interval width). While MVE and DNN outperformed XGBoost and achieved similar predictive accuracy, the MVE network provided stronger uncertainty quantification. It produced narrower PIs while maintaining valid coverage, which is critical for decision-making in power systems. The best model, a 5-model MVE ensemble, reached 95.3% coverage with an interval width of 0.1943, compared to the DNN's 95.0% coverage and 0.3074 width. This shows that MVE matches state-of-the-art accuracy while offering tighter and more reliable uncertainty bounds, reducing operational risk. Although the 5-model DNN ensemble's point-prediction metrics were strong (RMSE = 0.0939, MAE = 0.0469, R^2 = 0.8288), the wider intervals indicate higher underlying uncertainty, demonstrating that average accuracy alone can be misleading. Uncertainty-aware evaluation is therefore essential for reliable decision-making.

Appendix A
General Framework

The procedure followed in Section III is not specific to the case study and can be generalized to any one-dimensional regression problem with i.i.d. data. First, fit a regression model $\mu_\theta(\cdot)$ on the training set by minimizing MSE. Then train a neural network $\sigma_w^2(\cdot)$ to learn the uncertainty of the initial model in a Gaussian manner by minimizing the NLL

$$\mathcal{L}(w) = \frac{1}{2}\sum_i \log\left(2\pi\sigma_w^2(x_i)\right) + \sum_i \frac{(y_i - \mu_\theta(x_i))^2}{2\sigma_w^2(x_i)}. \quad (14)$$

Finally, use the trained pair $\left(\mu_\theta(\cdot), \sigma_w^2(\cdot)\right)$ to construct normalized residuals like Eq. 13, and apply split CP using a held-out calibration set.

References

[1] X.Chen *et al.*, "Electricity demand and grid impacts of ai data centers: Challenges and prospects," 2025.

[2] V. S.Rajkumar *et al.*, "Cyber attacks on power grids: Causes and propagation of cascading failures," *IEEE Access*, vol. 11, pp. 103 154–103 176, 2023.

[3] J.Gorka *et al.*, "Cascading blackout severity prediction with statistically-augmented graph neural networks," *Electric Power Systems Research*, vol. 234, p. 110738, 2024.

[4] R. A.Shuvro *et al.*, "Predicting cascading failures in power grids using machine learning algorithms," in *2019 North American Power Symposium (NAPS)*, 2019, pp. 1–6.

[5] M.Sun *et al.*, "Using bayesian deep learning to capture uncertainty for residential net load forecasting," *IEEE Transactions on Power Systems*, vol. 35, no. 1, pp. 188–201, 2019.

[6] Y.-H.Lin *et al.*, "A bayesian deep learning framework for rul prediction incorporating uncertainty quantification and calibration," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 10, pp. 7274–7284, 2022.

[7] Y.Gal *et al.*, "Concrete dropout," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 3584–3593.

[8] ——, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F.Balcan *et al.*, Eds., vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 1050–1059.

[9] H.Papadopoulos *et al.*, "Regression conformal prediction with nearest neighbours," *Journal of Artifical Intelligence Research*, vol. 40, no. 1, p. 815–840, Jan. 2011.

[10] P.Cui *et al.*, "Calibrated reliable regression using maximum mean discrepancy," 2020.

[11] D.-B.Wang *et al.*, "Rethinking calibration of deep neural networks: Do not be afraid of overconfidence," in *Advances in Neural Information Processing Systems*, M.Ranzato *et al.*, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 11 809–11 820.

[12] G.Shafer *et al.*, "A tutorial on conformal prediction," *Journal of Machine Learning Research*, vol. 9, p. 371–421, Jun. 2008.

[13] N. B.Nam *et al.*, "Comparative analysis of conformal prediction techniques and machine learning models for very short-term solar power forecasting," *Energy and AI*, vol. 21, p. 100573, 2025.

[14] B.Lakshminarayanan *et al.*, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[15] M.Noebels *et al.*, "Ac cascading failure model for resilience analysis in power networks," *IEEE Systems Journal*, vol. 16, no. 1, pp. 374–385, 2022.

[16] B. V.Venkatasubramanian *et al.*, "Machine learning based identification and mitigation of vulnerabilities in distribution systems against natural hazards," *IET Conference Proceedings*, vol. 2023, pp. 2908–2912, 2023.

[17] L.Sluijterman *et al.*, "Optimal training of mean variance estimation neural networks," *Neurocomputing*, vol. 597, p. 127929, 2024.

[18] T.Chen *et al.*, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 785–794.