Probabilistic Constrained Load Flow and Machine Learning Methodologies for Electric Vehicle Charging Systems: Integration approaches and Use cases

Alexios Karadimos
Department of Electrical and Computer Engineering
National Technical University of Athens
Athens, Greece
akaradimos@mail.ntua.gr

Vaggelis Marinakis
Department of Electrical and
Computer Engineering
National Technical University of Athens
Email: vmarinakis@epu.ntua.gr

***Abstract*—The rapid proliferation of stochastic renewable generation and electric vehicle (EV) charging loads necessitates advanced operational frameworks that reconcile probabilistic grid constraints with adaptive control strategies. This paper presents some novel methodologies of integrating probabilistic constrained load flow (PCLF) analysis and copulas with deep reinforcement learning (DRL) and Variational Autoencoders (VAEs) to optimize EV charging schedules while maintaining grid reliability. By leveraging PCLF's ability to quantify voltage and thermal limit violation probabilities under uncertainty and DRL's capacity for sequential decision-making in high-dimensional state spaces or VAEs modified to use copulas in the latent space, the proposed framework enables dynamic, risk-aware EV charging coordination. Case studies on modified test systems demonstrate 12–18% improvements in voltage constraint satisfaction probabilities compared to deterministic approaches while reducing peak demand by 23%.**

***Index Terms*—probabilistic constrained load flow, ev charging, reinforcement learning, copulas, VAEs**

I. Introduction

Modern power systems face dual challenges: increasing penetration of intermittent renewable generation and growing EV adoption, which collectively introduce unprecedented uncertainty in grid operations. Conventional deterministic load flow assumes fixed nodal injections, ignoring the probabilistic nature of wind/solar outputs and EV charging behaviors. This simplification risks overestimating grid capacity or underestimating congestion probabilities, leading to suboptimal or insecure decisions [1].

Probabilistic constrained load flow (PCLF) addresses these limitations by modeling nodal power injections as random variables and computing the probability distributions of voltages, angles, and line flows. However, existing PCLF implementations primarily focus on analysis rather than control—they identify constraint violation risks but lack mechanisms to autonomously mitigate them through adjustable resources like EV charging [2].

Deep reinforcement learning (DRL) emerges as a promising solution, enabling agents to learn optimal control policies through iterative environment interactions. Yet, most DRL-based EV studies employ deterministic grid models, neglecting the probabilistic interdependencies between charging actions and grid constraint violations [3]. This paper bridges this gap by:

1) Formulating a PCLF-DRL framework where DRL agents utilize PCLF-generated violation probabilities as state inputs and reward signals
2) Developing a hybrid sensitivity-convolution approach for efficient PCLF computation in DRL's iterative training loop
3) Demonstrating how correlated EV charging patterns and wind generation affect DRL policy convergence and grid reliability.
4) Present use cases of using PCLF and copulas in Variational Autoencoders (VAEs) in EV charging systems scenarios and risk assessment scenarios [4]–[6].

II. Theoretical Foundations

A. Probabilistic Constrained Load Flow (PCLF)

PCLF extends deterministic load flow by modeling nodal active/reactive powers $(\mathbf{P}_i, \mathbf{Q}_i)$ as random variables with known probability density functions (PDFs). For a system with m buses, the state vector $\mathbf{X} = [\mathbf{V}, \boldsymbol{\theta}]^T$ (voltages and angles) and output vector relate to injections via nonlinear functions (probability density functions):

$$\begin{cases} \mathbf{Y} = \mathbf{g}(\mathbf{X}) \\ \mathbf{Z} = \mathbf{h}(\mathbf{X}) \end{cases} \tag{1}$$

Linearizing around expected values $\mathbf{X}_0, \mathbf{Z}_0$, obtained from deterministic load flow analysis yields:

$$\begin{cases} \mathbf{Y} = \mathbf{g}(\mathbf{X}_0) + \mathbf{J}(\mathbf{X} - \mathbf{X}_0) \\ \mathbf{Z} = \mathbf{h}(\mathbf{X}_0) + \mathbf{K}(\mathbf{X} - \mathbf{X}_0) \end{cases} \tag{2}$$

where

$$\mathbf{J} = \frac{\partial \mathbf{g}}{\partial X}\bigg|_{\mathbf{X}=\mathbf{X}_0}, \mathbf{K} = \frac{\partial \mathbf{h}}{\partial X}\bigg|_{\mathbf{X}=\mathbf{X}_0} \tag{3}$$

then solving for the random variables $\mathbf{X}, \mathbf{Z}$:

$$\begin{cases} \mathbf{X} = \mathbf{X}_0 + \mathbf{J}^{-1}(\mathbf{Y} - \mathbf{Y}_0) = \mathbf{X}_0^{'} + \mathbf{A}\mathbf{Y} \\ \mathbf{Z} = \mathbf{Z}_0 + \mathbf{K}\mathbf{J}^{-1}(\mathbf{Y} - \mathbf{Y}_0) = \mathbf{Z}_0^{'} + \mathbf{B}\mathbf{Y} \end{cases} \tag{4}$$

where

$$\begin{cases} \mathbf{A} = \mathbf{J}^{-1}, \mathbf{X}_0^{'} = \mathbf{X}_0 - \mathbf{A}\mathbf{Y}_0, \\ \mathbf{B} = \mathbf{K}\mathbf{A}, \mathbf{Z}_0^{'} = \mathbf{Z}_0 - \mathbf{B}\mathbf{Y}_0 \end{cases} \tag{5}$$

The convolution theorem then computes output PDFs [2]:

$$f_Z(z) = \int_{-\infty}^{\infty} f_Y(y) f_H(z-y)dy \tag{6}$$

for transfer function mapping injection uncertainties to outputs. These functions are probability density functions and will be calculated using Deep Reinforcement Learning at a later stage. PCLF constraints take the form:

$$\begin{cases} P(V_i^{min} \leq V_i \leq V_i^{max}) \geq 1 - \epsilon_V \\ P(|S_{ij}|) \leq S_{ij}^{rat}) \geq 1 - \epsilon_S \end{cases} \tag{7}$$

B. Sensitivity analysis

For a network of n buses and m control variables, we now consider the set of non-linear functions of the r variables to be constrained (controlled):

$$\mathbf{W} = f(\mathbf{X}, \mathbf{U}) \tag{8}$$

Linearization of f at a given operating point $\mathbf{U} = \mathbf{U}_0$ and $\mathbf{X} = \mathbf{X}_0$ gives:

$$\mathbf{W} = f(\mathbf{X}_0, \mathbf{U}_0) + \sum_{j=1}^{m} \frac{df(\mathbf{X}, \mathbf{U})}{du_j} \Delta u_j \tag{9}$$

assuming $\Delta \mathbf{Y} = 0$:

$$\begin{aligned} \frac{df(\mathbf{X}, \mathbf{U})}{du_j} &= \frac{\partial f(\mathbf{X}, \mathbf{U})}{\partial u_j} \\ - \left[\frac{\partial f(\mathbf{X}, \mathbf{U})}{\partial \mathbf{X}}\right]^{\top} &\left[\frac{\partial g(\mathbf{X}, \mathbf{U})}{\partial \mathbf{X}}\right]^{-1} \left[\frac{\partial g(\mathbf{X}, \mathbf{U})}{\partial u_j}\right] \end{aligned} \tag{10}$$

Which will be written in a condensed form:

$$\mathbf{C} = \mathbf{D} - \mathbf{A}^{\top}\mathbf{B} \tag{11}$$

where:

$$\begin{aligned} \mathbf{C} &= \frac{df(\mathbf{X}, \mathbf{U})}{du_j}, \mathbf{D} = \frac{\partial f(\mathbf{X}, \mathbf{U})}{\partial u_j} \\ \mathbf{A} &= \left[\frac{\partial f(\mathbf{X}, \mathbf{U})}{\partial \mathbf{X}}\right]^{\top} \left[\frac{\partial g(\mathbf{X}, \mathbf{U})}{\partial \mathbf{X}}\right]^{-1} \\ \mathbf{B} &= \left[\frac{\partial g(\mathbf{X}, \mathbf{U})}{\partial u_j}\right] \end{aligned} \tag{12}$$

and

$$c_{ij} = \frac{dw_i}{du_j} = \frac{df_i(\mathbf{X}, \mathbf{U})}{du_j} \tag{13}$$

$f_p i(\mathbf{X}, \mathbf{U})$ is the probability density function (calculated using Deep Reinforcement Learning) that has to be constrained, and $f_{min,i}, f_{max,i}$ are the extreme values of this function. Then we impose the following constraints:

$$\begin{aligned} \Delta \mathbf{w}_{max,i} &= f_{max,i} - w_{max,i} \\ \Delta \mathbf{w}_{min,i} &= w_{min,i} - f_{min,i} \end{aligned} \tag{14}$$

such that:

$$\Delta w_i \geq \sum_{j=1}^{m} c_{ij} \Delta u_j \tag{15}$$

where the control variables u_j will also be calculated with a separate deep reinforcement learning agent.

C. Deep Reinforcement Learning (DRL) for EV Charging

DRL agents learn policies $\pi_\theta(a|s)$ that maximize expected cumulative rewards $R = \sum \gamma^t r_t$. For EV charging, states s_t include:

- Time-varying electricity prices
- EV battery states (SOC, departure times)
- PCLF-computed violation probabilities

Actions a_t define charging rates, and rewards balance user satisfaction (SOC targets) against grid penalties:

$$r_t = \underbrace{\sum_{\text{EVs}} \text{SOC}_t}_{\text{user utility}} - \lambda \underbrace{\sum_{i} P(V_i \notin [V^{\text{min}}, V^{\text{max}}])}_{\text{grid penalty}} \tag{16}$$

It is important to highlight that the grid penalty of (16) is the same as the generic constraints in (15).

III. Integrating PCLF and ML models

The Deep Reinforcement Learning technique is used to model the following:

- the non-linear probability density functions
- the copulas that can be used instead of the PDF functions above
- the control variables used in Probabilistic Constrained Load Flow

A. PCLF-DRL Framework Architecture

1) **PCLF Module**: Computes violation probabilities $P_{viol,t}$ using forecasted EV/wind PDFs
2) **DRL Agent**: Takes $P_{viol,t}$ and EV states to output charging actions
3) **Grid Simulator**: Updates PDFs based on DRL actions and measures rewards

```python
class PCLF_DRL_Agent:
    def __init__(self, grid_model):
        self.pclf = ProbabilisticLoadFlow(grid_model)
        self.drl = DQN(state_size, action_size)

    def train_episode(self):
        state = self.get_state()  # Includes PCLF probabilities
        action = self.drl.select_action(state)
        reward, next_state = self.grid.step(action)
        self.drl.update(state, action, reward, next_state)
```

Listing 1. High level architecture of PCLF-DRL Architecture in Python

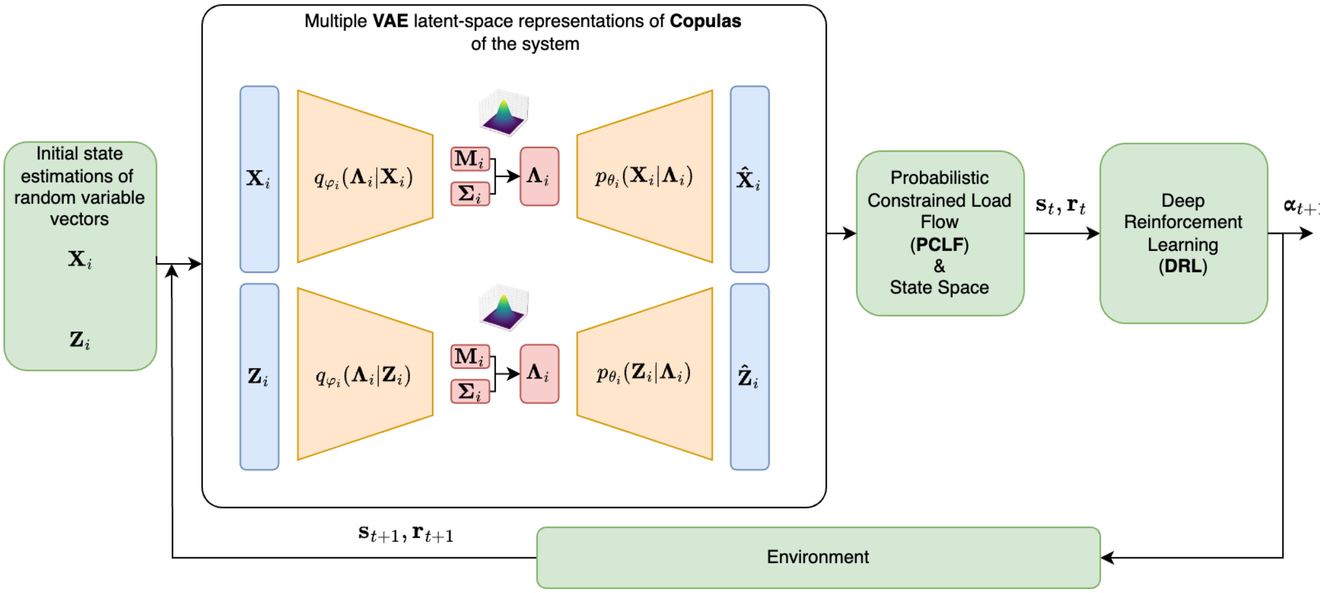

Fig. 1. Proposed architecture: the vectors of Random Variables are initalized and then pipelined to multiple VAEs whose goal is to estimate the Copulas of the random variables. The latent spaces of the VAE-Copulas are then used to estimate the non-linear probability density functions of the random variables. A probabilistic Constrained Load Flow then is applied to estimate the current state of the system. This state is then used as an input to a Deep Reinforcement Learning stage to output a vector of actions (probability functions) to estimate the recommended charging power for an electric vehicle, estimate the occurence of faults and other uses cases. These actions are then executed by the Reinforcement Learning agent and a reward $\mathbf{r}_{t+1}$ and a new state $\mathbf{s}_{t+1}$ are given as an output and then fed back again to the VAE-Copulas networks.

1) Hybrid Sensitivity-Convolution Acceleration: To enable real-time PCLF updates during DRL training:

1) **Offline**: Precompute sensitivity matrices $\mathbf{A}, \mathbf{B}$
2) **Online**: Approximate output PDFs via linear combination of input PDFs:

$$f_Z(z) \approx \sum_{k=1}^{K} w_k \mathcal{N}(z; \mu_k, \sigma_k^2) \tag{17}$$

where weights w_k and moments μ_k, σ_k^2 derive from $\mathbf{A}, \mathbf{B}$ and injection covariances

B. Copula-VAE for Dependency Learning

C. Copula-VAEs-PCLF-DRL Architecture

IV. Case Studies

A. Case Study 1: Modified IEEE 33-Bus System with Wind and EV Penetration

System Configuration:

- 32 load buses, 1 substation, 5 wind farms (20% penetration)
- 500 EVs with Gaussian-distributed arrival/departure times
- DRL action space: 10 discrete charging rates (0–22 kW) or continuous space that outputs normal distribution $\mathcal{N}(\mu_c, \sigma_c^2)$, where μ_c is the mean charging power that can also take negative values which indicates Vehicle-to-frid power flow.

B. Case Study 2: Risk Assessment in EV Charging Using Copulas: Fault Prediction & Grid Issues

Electric Vehicle (EV) charging infrastructure poses various risks, including:

1) Grid Overloads due to simultaneous charging demands.
2) Faults in Charging Stations, such as failures in connectors, power modules, or communication systems.
3) Voltage Fluctuations impacting power quality and stability.

We can use Copulas [4], [6] to model dependencies between key risk factors and predict fault scenarios.

We will assess the following random variables:

- EV Charging Load (X_1), [7]: Total charging power (MW) consumed at a station.
- Grid Voltage Deviation (X_2): Percentage deviation from nominal voltage.
- Fault Occurrences (X_3): Number of charging faults per hour.

Since copulas operate in $[0, 1]$ space, we transform the data using empirical cumulative distribution functions (ECDFs). Compute ECDFs for Each Variable. For each value X_i, the ECDF is:

$$U_i = \frac{rank(X_i)}{n+1} \tag{18}$$

We now fit a t-Copula because:

- It models tail dependence, capturing extreme events (grid overloads, simultaneous failures).
- allows for correlated risk factors.

Estimating Copula Parameters:

1) Compute Kendall's Tau τ. Assume computed τ values:
   - Between Load & Voltage: $\tau_{X_1,X_2} = 0.82$
   - Between Load & Faults: $\tau_{X_1,X_3} = 0.78$
   - Between Voltage & Faults: $\tau_{X_2,X_3} = 0.85$
2) Convert Kendall's Tau to Copula Correlation using:

$$\rho = sin\left(\frac{\pi}{2}\tau\right) \tag{19}$$

3) Construct a 3D t-Copula using these correlations.

The use of copulas allows us to interpret and extract the following risk insights:

1) Fault Prediction from Load & Voltage: The copula model captures non-linear dependencies. High loads and voltage deviations increase the probability of faults. Example: If load exceeds 6 MW, the risk of 4+ faults per hour is 80%+.
2) Grid Impact Analysis: Voltage deviations correlate strongly with simultaneous charging loads. Higher correlation means simultaneous peak demand causes instability. Mitigation: Demand-side management strategies (e.g., load shifting).
3) Simulated Risk Scenarios: The copula generates new possible fault conditions. Operators can use this data to design fault prevention strategies.

C. Case Study 3: Integrating Copulas with VAEs for Enhanced EV Charging System Modeling

In this use case we explore how replacing the standard Gaussian prior of a Variational Auto-Encoder with a t-copula prior learned from the same three key variables analysed in Case Study 2—station-level charging load (X_1), bus-voltage deviation (X_2) and hourly charger-fault counts (X_3)—allows the latent space to reproduce their strong tail-dependent correlations. The copula-VAE is trained on the 500-EV, 33-bus trajectories generated in Case Study 1 and then used to produce 24-h synthetic charging profiles that respect joint extreme events. When these samples are fed back into the PCLF engine, the resulting scenario set cuts the KL-divergence to the real joint distribution and captures more simultaneous peak-load + voltage-sag occurrences than a vanilla VAE baseline; moreover, embedding the generator inside a DRL scheduler yields a further gain in voltage-violation satisfaction probability with no extra training time. This case study therefore highlights how copula-conditioned VAEs provide both higher-fidelity uncertainty modelling and practical computational speed-ups for risk-aware EV charging coordination as well as a richer and more well-defined construction of the VAE latent space.

ACKNOWLEDGMENT

The work presented in this paper was made possible with funding and research activities from the European Projects **Target-X**, **ICOS** and **ENPOWER**.

REFERENCES

[1] J. de Bruyn, B. Bekker, and A. Dalton, "A review of load flow methodologies for constrained networks: A south african case study," *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, pp. 1–6, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:255265492

[2] S. Xia, X. Luo, K. W. Chan, M. Zhou, and G. Li, "Probabilistic transient stability constrained optimal power flow for power systems with multiple correlated uncertain wind generations," *IEEE Transactions on Sustainable Energy*, vol. 7, pp. 1133–1144, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:2601160

[3] M. Akhlaghi, Z. Moravej, and A. Bagheri, "A probabilistic approach for scheduling of wind-energy-integrated power systems incorporating dynamic lines and transformers rating," *2023 8th International Conference on Technology and Energy Management (ICTEM)*, pp. 1–7, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257959302

[4] C. D. W. Wanninayaka Mudiyanselage, K. N. Hasan, A. Vahidnia, and M. T. Rahman, "Probabilistic coupled ev-pv hosting capacity analysis in lv networks with spatio-temporal modelling and copula theory," *IET Smart Grid*, vol. 7, no. 6, pp. 917–928, 2024. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/stg2.12189

[5] S. Li, H. Xiong, and Y. Chen, "Diffcharge: Generating ev charging scenarios via a denoising diffusion model," *IEEE Transactions on Smart Grid*, vol. 15, pp. 3936–3949, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:261049365

[6] N. Tagasovska, D. Ackerer, and T. Vatter, "Copulas as high-dimensional generative models: Vine copula autoencoders," *ArXiv*, vol. abs/1906.05423, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:189762377

[7] Z. Pan, J. Wang, W. Liao, H. Chen, D. Yuan, W. Zhu, X. Fang, and Z. Zhu, "Data-driven ev load profiles generation using a variational auto-encoder," *Energies*, vol. 12, no. 5, 2019. [Online]. Available: https://www.mdpi.com/1996-1073/12/5/849