UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL-REI - UFSJ
Departamento de Engenharia Elétrica - DEPEL

AVALIAÇÃO DA CONFIABILIDADE DE SISTEMAS ELÉTRICOS DE POTÊNCIA POR MEIO DE TÉCNICAS DE APRENDIZADO POR REFORÇO

Daniel Maia de Guadalupe
Discente do curso de Engenharia Elétrica

Prof. Dr. Fernando Aparecido de Assis
Departamento de Engenharia Elétrica

São João del-Rei
Setembro de 2025

RESUMO

A avaliação da confiabilidade composta em Sistemas Elétricos de Potência (SEP) tradicionalmente é realizada por métodos baseados em simulações de Monte Carlo associadas ao Fluxo de Potência Ótimo (FPO). Embora essa abordagem ofereça alta precisão, ela possui um elevado custo computacional, o que dificulta a realização de avaliações em tempo real e em sistemas de grande escala. Nesse sentido, o presente trabalho investiga uma abordagem alternativa para realizar avaliações de confiabilidade baseada em Aprendizagem por Reforço (AR). A proposta é treinar um agente para atuar como um operador autônomo do sistema, tomando decisões de despacho de geradores e corte de carga após a ocorrência de contingências. Com isso, busca-se superar a limitação computacional do método clássico de avaliação, possibilitando a estimativa de índices de confiabilidade com maior eficiência computacional. Apesar de os resultados obtidos não terem atingido o desempenho esperado, o estudo detalha a metodologia implementada, os ambientes de simulação desenvolvidos, os resultados obtidos e os desafios encontrados, contribuindo para o debate sobre aplicações de AR em confiabilidade de SEP e apontando direções para pesquisas futuras.

Palavras-chave: Aprendizagem por Reforço, Confiabilidade, Sistemas Elétricos de Potência.

1 INTRODUÇÃO

A eletricidade é um pilar fundamental para a sociedade contemporânea. A importância desse recurso confere ao SEP um papel estratégico, o que exige que sua operação entregue níveis elevados de confiabilidade e de disponibilidade. No entanto, esse sistema é composto por uma ampla rede de componentes interconectados, o que o torna suscetível a falhas de caráter estocástico, decorrentes tanto de limitações de vida útil dos equipamentos quanto de eventos externos.

Na rede básica de transmissão do Brasil, a confiabilidade é assegurada pelo critério determinístico N-1, definido pelos Procedimentos de Rede do Operador Nacional do Sitema Elétrico [1], que exige que o sistema suporte a perda de um componente sem comprometer sua operação segura. Embora eficaz, esse método não considera a importância relativa dos equipamentos para a confiabilidade do sistema e pode levar a uma alocação ineficiente de recursos. Por esse motivo, a avaliação probabilística da confiabilidade composta é aplicada de forma complementar, baseada em Simulações de Monte Carlo (SMC), possiblitando a estimativa de indicadores de confiabilidade do sistema. No entanto, o alto custo computacional envolvido nessas simulações limita sua aplicação.

Diante desse desafio, as técnicas de *Machine Learning* (ML) têm se consolidado como alternativas promissoras para otimizar as avaliações de confiabilidade composta. Na maioria das aplicações, os modelos de ML são usados para pré-classificar os estados amostrados do sistema em duas categorias: sucesso ou falha, de acordo com a probabilidade de ocorrência de corte de carga. Essa estratégia reduz a necessidade de execuções do FPO apenas aos casos críticos, trazendo ganhos significativos. Nesse contexto, os métodos de aprendizagem supervisionada têm sido os mais empregados na literatura [2], embora existam exemplos de aplicação de métodos não supervisionados, como em [3].

Este trabalho investiga a aplicação da AR ao problema da avaliação da confiabilidade composta de SEPs. A AR é um ramo do ML no qual o modelo aprende por meio da interação entre um agente e o ambiente, sem a necessidade de bases de dados pré-existentes. No contexto de confiabilidade de SEPs, essa abordagem foi explorada de forma pioneira por [4], que treinou um agente capaz de atuar de maneira similar a um operador humano, tomando decisões de despacho de geradores e de cortes de carga após falhas. O objetivo do agente é restaurar o sistema a uma condição onde não há violação de limites operacionais, priorizando a minimização do corte de carga e o despacho econômico da geração. A partir desse estado controlado, são estimados os índices de confiabilidade.

Nesse sentido, este relatório descreve as estratégias e adaptações realizadas buscando reproduzir o estudo mencionado. O treinamento do agente é avaliado em ambientes de Simulação de Monte Carlo Sequencial (SMC-S) e Não Sequencial (SMC-NS), com o objetivo de contribuir para a discussão sobre o uso de técnicas de AR na avaliação da confiabilidade de SEPs.

2 CONCEITOS PRELIMINARES

2.1 Avaliação da Confiabilidade Composta de Sistemas Elétricos de Potência

A avaliação da confiabilidade composta analisa a capacidade de um SEP de atender integralmente à demanda de potência sem violar limites operacionais, levando em conta a possibilidade de falhas em unidades geradoras e elementos da rede de transmissão. Por se tratar de uma abordagem probabilística, o desempenho do sistema é mensurado por índices de confiabilidade associados a eventos de corte de carga.

Os índices de confiabilidade são obtidos por simulações que avaliam a adequação estática de diferentes estados do sistema, definidos pela combinação das disponibilidades dos componentes conforme suas distribuições de probabilidade de falha. Na avaliação da confiabilidade composta de SEPs, duas variantes principais da SMC são empregadas [5]. A SMC-NS gera estados independentes a partir das distribuições de falha e reparo e permite estimar índices estáticos como LOLP (*Loss of Load Probability*), LOLE (*Loss of Load Expectation*), EPNS (*Expected Power Not Supplied*) e EENS (*Expected Energy Not Supplied*). Já a SMC-S simula a evolução cronológica do sistema, incorporando a dependência temporal dos eventos; além dos índices citados para a SMC-NS, possibilita estimar índices relacionados à cronologia das falhas, como a duração e a frequência dos cortes de carga. Enquanto a SMC-NS tende a ser menos onerosa computacionalmente, a SMC-S apresenta resultados mais realistas por representar explicitamente a cronologia das falhas e reparos dos equipamentos.

A estimativa dos índices de confiabilidade é feita por meio de funções de teste, que registram variáveis de interesse ao longo da simulação de um SEP. A cada iteração, um estado x do sistema é avaliado em termos de adequação estática, utilizando um algoritmo de fluxo de potência com medidas corretivas, como o FPO linearizado ou o FPO AC. A partir do resultado, as funções teste $F(x)$ para os índices de confiabilidade que se deseja estimar são calculadas e os estimadores são atualizados utilizando a Equação (2.1). A título de exemplo, a função de teste do índice EPNS é apresentada na Equação e (2.2), onde ΔP, em MW, representa o corte de carga realizado em estados de falha. Maiores detalhes sobre os índices de confiabilidade são apresentados por [5].

$$\tilde{E}[F] = \frac{1}{N_S} \sum_{k=1}^{N_S} F_{\tilde{E}[F]}(x) \tag{2.1}$$

$$F_{\text{EPNS}}(x) = \begin{cases} \Delta P, & \text{se o estado é de falha} \\ 0, & \text{caso contrário} \end{cases} \tag{2.2}$$

O coeficiente de variação relacionado à variância do estimador $V(\tilde{E}[F])$, utilizado para veri-

ficação da convergência do algoritmo, é determinado como:

$$\beta_{\tilde{E}[F]} = \sqrt{V(\tilde{E}[F])/\tilde{E}[F]} \tag{2.3}$$

O processo iterativo é finalizado quando o coeficiente de variação β se torna inferior ao limite de tolerância previamente estabelecido para a variância do estimador [5].

2.2 Aprendizagem por Reforço

A AR é um ramo do ML voltado para a solução de problemas de decisão sequencial. Nessa abordagem, um agente interage continuamente com um ambiente: observa estados, seleciona e executa ações e recebe recompensas como consequência. Esse ciclo de interação permite que o agente ajuste gradualmente sua forma de agir de acordo com os retornos obtidos. O objetivo principal é que o agente aprenda uma política π, ou seja, um conjunto de regras de decisão capaz de maximizar a soma das recompensas acumuladas ao longo dos episódios [6].

2.2.1 Processo de Decisão de Markov (PDM)

A interação entre agente e ambiente é formalmente modelada por um PDM, uma estrutura matemática canônica empregada em problemas de decisão sequencial [6]. O que define um PDM é a Propriedade de Markov, que determina que a evolução do sistema depende exclusivamente do estado presente e da ação tomada, sem levar em conta o histórico completo [6]. Essa premissa é amplamente adotada no desenvolvimento de algoritmos de aprendizagem por reforço, pois simplifica o problema ao dispensar a análise de toda a trajetória de estados e ações. A partir dessa formulação, são definidas as funções de valor e as equações de Bellman [7], ferramentas fundamentais para avaliar estados e orientar a busca por políticas ótimas em algoritmos de aprendizagem por reforço.

2.2.2 Funções de Valor e Equações de Bellman

Para avaliar a qualidade de uma política, os algoritmos de RL empregam funções de valor que associam a cada estado ou par estado-ação o retorno esperado [6].

A função valor de estado ($V^{\pi}(s)$) é definida como o retorno esperado acumulado ao se tomar a ação *a* no estado *s* e, posteriormente, seguir a política π [6]. Essa função é definida na Equação 2.4:

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, s_0 = s\right], \tag{2.4}$$

Na equação, r_t representa a recompensa recebida no instante t, enquanto γ representa um fator de desconto. Esse fator pondera a importância de recompensas futuras. Assim, quando $\gamma = 1$, todas

as recompensas, independentemente de quando são recebidas, têm o mesmo peso. À medida que γ é reduzido, a função de valor passa a priorizar as recompensas imediatas.

De forma análoga, a função valor-ação ($Q^{\pi}(s,a)$) é definida como o retorno esperado acumulado ao se tomar a ação *a* no estado *s* e, posteriormente, seguir a política π [6]. Essa função é definida na Equação 2.5.

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^{t} r_t \,\middle|\, s_0 = s, a_0 = a\right]. \tag{2.5}$$

A Equação de Bellman [7] estabelece a relação entre o valor de um estado e o valor dos estados subsequentes. Dessa forma, ela permite decompor o valor de um estado na soma de duas parcelas mais simples: a recompensa imediata e o valor dos estados futuros. Essa função é definida na Equação 2.6.

$$Q^{\pi}(s,a) = \mathbb{E}\left[r_t + \gamma\, Q^{\pi}(s_{t+1}, a_{t+1}) \mid s_t = s, a_t = a\right] \tag{2.6}$$

Para o caso ótimo, busca-se a política π^{*} que maximize o retorno esperado. A equação de otimalidade de Bellman [7] para a função valor-ação é expressa pela Equação 2.7:

$$Q^{*}(s,a) = R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) \max_{a'} Q^{*}(s',a'). \tag{2.7}$$

2.2.3 Algoritmos de Aprendizagem por Reforço

A aplicação das formulações apresentadas na Seção 2.2.2 possibilitou a construção de algoritmos canônicos da AR, como o Q-Learning [8] e o SARSA [9]. Esses algoritmos, contudo, foram desenvolvidos como métodos tabulares, armazenando uma tabela explícita de valores $Q(s,a)$ para cada par estado-ação. Esta abordagem impõe limitações severas em problemas com espaços de estados e ações grandes ou contínuos, como é o caso de aplicações em SEP, devido à maldição da dimensionalidade [7] e à incapacidade de generalização do agente para estados não visitados.

Para superar essas limitações, houve, nas últimas décadas, uma convergência do *framework* clássico de AR com os avanços no aprendizado de máquina, dando origem à Aprendizagem por Reforço Profunda (ARP) [6, 10]. Neste novo paradigma, redes neurais profundas são utilizadas como aproximadores de função para representar políticas $\pi(s)$ ou funções valor $V(s)$ e $Q(s,a)$, permitindo generalizar a partir de experiências passadas do agente e lidar com espaços de alta dimensionalidade.

3 METODOLOGIA

Este capítulo apresenta a metodologia adotada para a implementação dos ambientes de treinamento de agentes de AR para a avaliação de confiabilidade de SEPs. São descritos os recursos computacionais, as bibliotecas utilizadas e os sistemas-teste escolhidos, bem como os ajustes metodológicos realizados ao longo do trabalho, motivados pelas dificuldades observadas no processo de treinamento dos agentes.

3.1 Ferramentas e recursos computacionais empregados

Os ambientes de simulação foram desenvolvidos utilizando a biblioteca OpenAI Gymnasium, que fornece uma interface padronizada para a definição de estados, ações e recompensas em problemas de tomada de decisão sequencial. O treinamento dos agentes foi realizado com o uso da biblioteca Stable Baselines 3, que disponibiliza implementações estáveis e otimizadas de algoritmos de DRL. O monitoramento de métricas ao longo do processo de aprendizado foi realizado por meio do TensorBoard, uma ferramenta usada para visualização em tempo real de variáveis em aplicações de ML. Todas as etapas foram implementadas em Python 3 e executadas na plataforma Google Colab, aproveitando a aceleração do processo de treinamento proporcionada pela GPU Tesla T4.

3.2 Modelo de AR adotado

O modelo de AR adotado neste trabalho foi o *Twin Delayed Deep Deterministic Policy Gradient* (TD3) [11], um algoritmo projetado para operar em ambientes com espaço de ação contínuo, como é o caso do SEP. O TD3 combina três estratégias principais para melhorar a estabilidade e eficiência do treinamento: (i) dois críticos independentes para mitigar a superestimação dos valores de Q; (ii) atualização atrasada da política em relação aos críticos, reduzindo a propagação de erros; e (iii) ruído nas ações-alvo para suavizar o aprendizado. Além disso, foram utilizados o otimizador Adam e a técnica de *dropout*, visando obter maior robustez e maior capacidade de generalização do modelo.

3.3 Sistema de teste

Para a avaliação do desempenho dos agentes, foi empregado um sistema de teste amplamente utilizado em estudos de confiabilidade de SEP, o IEEE Reliability Test System (IEEE RTS-24) [12]. Esse sistema é composto por 24 barras, das quais 10 correspondem a barras de geração com um total de 32 unidades geradoras, e 17 correspondem a barras de carga, interconectadas por 38 linhas de transmissão. Esse sistema possui capacidade instalada de 3405 MW e uma carga máxima de 2850

MW. O sistema adota uma curva horária determinística para o perfil de carga, sendo o ajuste aplicado de forma uniforme a todas as barras.

3.4 Ambiente de avaliação de confiabilidade via MCS-S

O ambiente de treinamento para o agente de avaliação de confiabilidade composta foi implementado conforme a metodologia proposta por [4]. Nesta simulação, o agente tem como objetivo minimizar o custo operativo do sistema, controlando o despacho de geradores nas barras PV e executando o corte de carga quando necessário. Essas ações são escolhidas buscando evitar sobrecargas térmicas nas linhas de transmissão e assegurar que a geração na barra *swing* permaneça dentro dos seus limites físicos.

A interação do agente com o ambiente é ilustrada no fluxograma da Figura 1. A cada passo de tempo t, o agente recebe o estado atual s_t do ambiente. Com base nesse estado, a política π calcula o vetor de ações a_t, que é aplicado ao sistema de potência. Em seguida, resolvem-se as equações de fluxo de potência para determinar um estado intermediário do sistema. A partir desse estado, calcula-se a recompensa r_t. O sistema é então avançado no tempo, incorporando variações de carga ou mudanças no estado dos componentes, e as equações de fluxo de potência são novamente resolvidas, obtendo-se o novo estado s_{t+1}, a partir do qual é gerada uma nova observação que é repassada ao agente. As experiências do agente $\{s_t, a_t, r_t, s_{t+1}\}$ são armazenadas em um *replay buffer*, a partir do qual o otimizador de política amostra lotes de dados para atualizar a política, de modo que $\pi \rightarrow \pi'$.

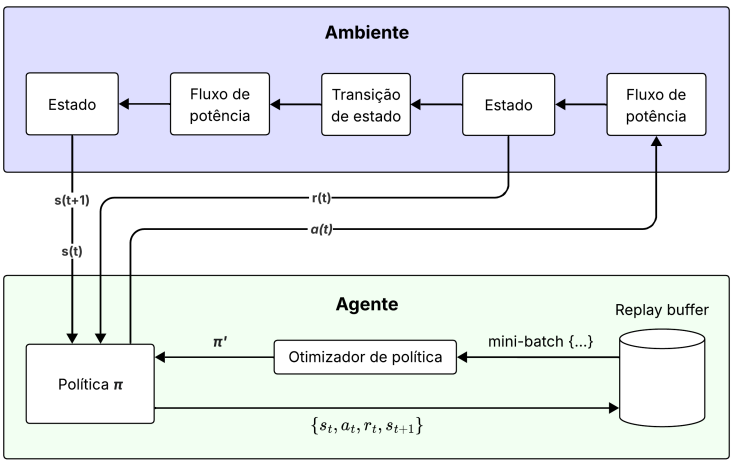

Figura 1 – Interação do agente com o ambiente
Diagrama adaptado de [4].

Para a simulação do sistema elétrico neste trabalho, empregou-se a formulação de fluxo de potência DC (FPDC). Essa abordagem é a mais empregada para aplicações de larga escala e que não exigem precisão elevada, como é o caso do treinamento de agentes de aprendizagem por reforço, que requerem a execução de milhões de iterações. É válido destacar que uma única interação do agente

com o ambiente demanda duas execuções de fluxo de potência, o que torna a eficiência computacional um fator ainda mais crítico para essa aplicação.

A formulação DC é obtida a partir das hipóteses de simplificação usuais: as resistências das linhas são desprezadas, assume-se pequenas diferenças angulares entre barras vizinhas e as tensões são consideradas unitárias em todas as barras. Essas simplificações permitem linearizar o problema, eliminando a necessidade de métodos iterativos para determinação do estado operativo do sistema.

A estrutura do ambiente é descrita nas subseções seguintes.

3.4.1 Inicialização do ambiente

Para representar o problema como um PDM, o tempo é discretizado pela divisão de cada hora em $N_h = 12$ intervalos de tamanho fixo δt, de modo que o agente interage com o ambiente a cada 5 minutos. O treinamento nesse ambiente ocorre em uma SMC-S, no qual as falhas dos componentes são pré-amostradas para todo o período de simulação, gerando uma linha temporal de eventos. O estado inicial de cada componente é sorteado com probabilidade $p^c_{\text{up}} = \lambda^c_r/(\lambda^c_r + \lambda^c_f)$, onde λ^c_r é a taxa de reparo e λ^c_f é a taxa de falha de um componente. As durações de operação e falha são amostradas alternadamente a partir de distribuições exponenciais dos parâmetros λ^c_f e λ^c_r. Como o tempo é discretizado, cada evento contínuo é associado ao intervalo discreto mais próximo.

3.4.2 Definição dos episódios

O ambiente de treinamento é definido de forma episódica e com recompensas negativas para incentivar que o agente finalize o episódio o mais rápido possível. Cada episódio parte do estado atual do sistema e é dividido em duas fases: operação segura e pós-contingência, iniciada quando ocorrem falhas de componetes ou quando o consumo se torna maior que a geração. O episódio é encerrado ao atingir um número máximo de passos N_{max}, definido como 255 passos, ou quando o agente restaura o sistema a uma condição segura, ou seja: sem sobrecargas, cortes de carga ou violações nos limites da barra *swing*.

3.4.3 Espaço de ações

A cada passo de tempo, o agente pode ajustar a potência dos geradores que não estão na barra *swing* ou realizar cortes de carga. Essas ações são representadas da seguinte forma:

- **Ajuste na geração:** Ajustar para cima ou para baixo a potência de saída de estações geradoras que estejam em operação e eletricamente conectadas à barra *swing*, mas não localizadas nela. Todos os geradores de uma estação são ajustados de forma proporcional, de forma que o controle de geração é feito a nível de barra.

• **Corte de carga:** Reduzir ou restaurar a potência demandada em cada uma das barras de carga que estejam eletricamente conectadas à barra *swing*.

Todas as cargas que, devido à eventual indisponibilidade da rede de transmissão, não estejam eletricamente conectadas à barra *swing* são consideradas inteiramente cortadas. Para esta aplicação, foram avaliadas duas estratégias de aplicação das ações: a abordagem incremental, na qual um valor de potência positivo ou negativo é adicionado ao valor de potência atual; e a abordagem proporcional, na qual a potência das unidades geradoras e das cargas é selecionada livremente como uma porcentagem do total disponível, variando entre 0% e 100%.

3.4.4 Espaço de observação

A cada passo de tempo, uma observação do ambiente é repassada ao agente. O vetor de observação é composto pelos seguintes elementos:

1. Fase do episódio;
2. Capacidade relativa máxima disponível na barra *swing*;
3. Capacidade relativa mínima disponível na barra *swing*;
4. Despacho das estações geradoras controláveis;
5. Capacidades máximas disponíveis das estações geradoras;
6. Corte de carga por barra;
7. Cargas nominais conectadas;
8. Carregamento relativo transformado das linhas de transmissão;
9. Estados das linhas de transmissão.

3.4.5 Função recompensa

A estrutura de recompensas do ambiente é construída de forma a refletir o que se deseja que o agente aprenda. Do ponto de vista operacional, o objetivo é reduzir os custos associados a falhas no sistema, priorizando o corte das cargas de menor valor e mantendo as interrupções com a menor duração possível. Para isso, a cada passo de tempo o agente recebe uma recompensa negativa proporcional ao custo do corte de carga e ao custo de geração, além de penalidades adicionais por violações em linhas de transmissão e na barra *swing*.

O custo de geração é definido pela Equação 3.1:

$$Custo_{ger} = \sum_{g \in G} c_{ger}(P_g) \tag{3.1}$$

O custo de corte é definido pela Equação 3.2, onde c_{ger} e c_{corte} são funções de mapeamento entre o despacho do gerador ou o corte de carga e o respectivo custo.

$$Custo_{corte} = \sum_{g \in G_D} c_{corte}(P_g) \tag{3.2}$$

Assim, o custo total de produção é dado pela Equação 3.3:

$$C_{prod} = Custo_{ger} + Custo_{corte} \tag{3.3}$$

O custo de sobrecarga das linhas de transmissão é modelado pela Equação 3.4, onde k_l é uma constante positiva que pondera a penalidade associada a cada linha.

$$C_{linhas} = \sum_{l \in L} k_l \cdot \max\{0, |P_l^{fluxo}| - P_{fluxo,l}^{\max}\} \tag{3.4}$$

O custo associado à barra *swing* é definido na Equação 3.5, em que k_{sw} é uma constante positiva, $P_{sw}^{\min}$ e $P_{sw}^{\max}$ são os limites nominais inferior e superior, respectivamente, e P_{sw} é a geração ativa na barra *swing*.

$$C_{sw} = k_{sw} \cdot \max\{0, P_{sw}^{\min} - P_{sw},\ P_{sw} - P_{sw}^{\max}\} \tag{3.5}$$

Por fim, a recompensa em cada passo de tempo é dada pela Equação 3.6, onde k_{rec} é uma constante positiva de normalização, utilizada para melhorar a estabilidade do treinamento, mantendo os valores de recompensa dentro de uma faixa adequada.

$$r = \frac{-C_{prod} - C_{line} - C_{sw}}{k_{rec}} \tag{3.6}$$

3.4.6 Cronologia

As transições do ambiente do sistema seguem a seguinte sequência lógica:

1. Inicializar o episódio.

2. Se ocorrerem alterações no estado do sistema, calcular o estado atual por meio da solução das equações de fluxo de potência.

3. Atualizar limites de geração e lista de barras conectadas eletricamente à barra *swing* com base na disponibilidade de componentes do sistema.

4. Calcular e aplicar as ações e, em seguida, calcular o estado intermediário do sistema.

5. Calcular a recompensa.

6. Aplicar quaisquer alterações estocásticas nos estados dos componentes ou nas cargas.

7. Avançar no tempo: $t = t + \delta t$. Se for o fim do episódio, retornar ao passo 0; caso contrário, retornar ao passo 1.

3.5 Ambiente de avaliação de confiabilidade via MCS-NS

Diante da dificuldade em obter resultados consistentes com a aplicação do *framework* de aprendizagem por reforço em um ambiente sequencial de avaliação de confiabilidade, optou-se por investigar uma simplificação do problema por meio da implementação do treinamento em uma MCS-NS. Essa abordagem preserva a estrutura principal do ambiente original, diferenciando-se apenas na forma de amostragem dos estados e na definição dos episódios.

Enquanto no ambiente sequencial os episódios eram definidos a partir de uma linha temporal, no ambiente não sequencial o sistema é reinicializado ao final de cada episódio, sendo amostrado um novo estado independente. Essa modificação removeu a dimensão temporal do problema, transformando-o em uma série de instantes independentes. A hipótese que motivou essa transição foi a de que a definição pouco clara dos episódios no ambiente sequencial representava um obstáculo adicional à aprendizagem. Na abordagem não sequencial, a definição de episódio é mais simples: o agente interage com um único estado do sistema até que encontre uma solução ou esgote um limite de ações.

3.6 Parâmetros do ambiente e hiperparâmetros das redes neurais

Em ambos os ambientes testados, todos os parâmetros e hiperparâmetros foram mantidos conforme o estudo de [4], definidos nas Tabelas I e II, de modo a evitar que a escolha dos hiperparâmetros se tornasse um fator limitante para a aprendizagem do agente. A única modificação foi a redução do número total de passos de treinamento, de 14.500.000 para 500.000, em razão das restrições computacionais disponíveis. Esse ajuste foi considerado na análise dos resultados.

3.7 Variáveis monitoradas

A fim de avaliar o desempenho do agente ao longo do treinamento, foram monitoradas as seguintes variáveis:

- Recompensa média por episódio;
- Duração média dos episódios;
- Corte percentual da carga total do sistema;
- Potência gerada na barra *swing*;
- Custo por violações na barra *swing*;
- Custo por violações nas capacidades das linhas de transmissão.

4 RESULTADOS

Neste capítulo, são apresentados os resultados obtidos a partir das simulações realizadas nos ambientes descritos no Capítulo 3. Os testes foram conduzidos ao longo de 500.000 iterações, das quais 100.000 corresponderam a uma fase de aquecimento do agente, na qual as ações são escolhidas de forma aleatória com o objetivo de preencher o *replay buffer*. Embora uma quantidade maior de iterações totais pudesse fornecer uma estimativa mais precisa do desempenho do agente, limitações de tempo de GPU da plataforma Google Colab restringiram a extensão dos treinamentos.

4.1 Ambiente de confiabilidade via SMC-S - Ações incrementais

Os primeiros experimentos foram conduzidos no ambiente sequencial (SMC-S) aplicado ao sistema IEEE-RTS de 28 barras, aplicando as ações de forma incremental.

O desempenho do agente pode ser avaliado inicialmente pela evolução da duração média dos episódios. A partir da Figura 2, observa-se que, após o início do treinamento, o agente aprende rapidamente a reduzir a duração dos episódios, passando a oscilar em torno de 10 passos por episódio. Após os 250.000 passos, contudo, há um aumento repentino na duração dos episódios, demonstrando instabilidade no treinamento. Essas oscilações de controle são comuns durante treinamentos de agentes RL, mas também podem evidenciar falhas no *design* do ambiente.

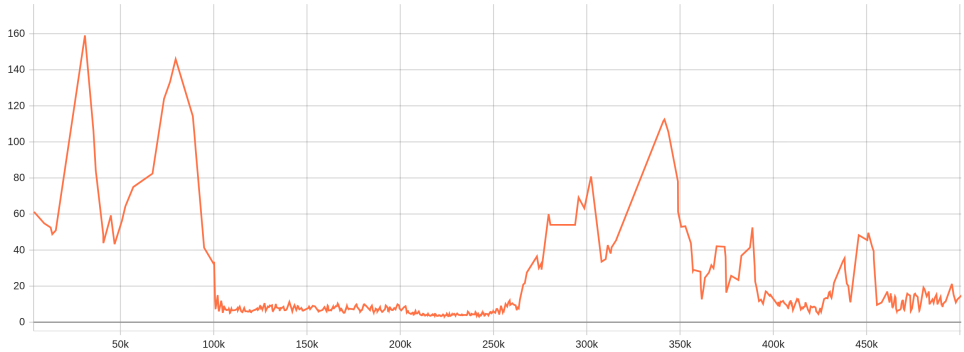

Figura 2 – Duração média dos episódios - SMC-S com ações incrementais

A evolução da recompensa média por episódio é mostrada na Figura 3. Observa-se um ganho expressivo logo após a fase de aquecimento, quando tem início o treinamento das redes neurais. A recompensa média alcança o valor máximo de -3,75 próximo aos 200.000 passos, mas a partir desse ponto passa a apresentar instabilidade e, mesmo quando se estabiliza, não apresentam uma melhora relevante. Isso sugere que o agente aprendeu a evitar punições severas, mas não conseguiu definir uma estratégia capaz de maximizar a recompensa. Essa limitação pode estar relacionada a um baixo comportamento exploratório por parte do agente. Para investigar essa hipótese, aumentou-se o ruído

de ação, que é o mecanismo usado no algoritmo TD3 para promover a exploração, mas os resultados permaneceram semelhantes.

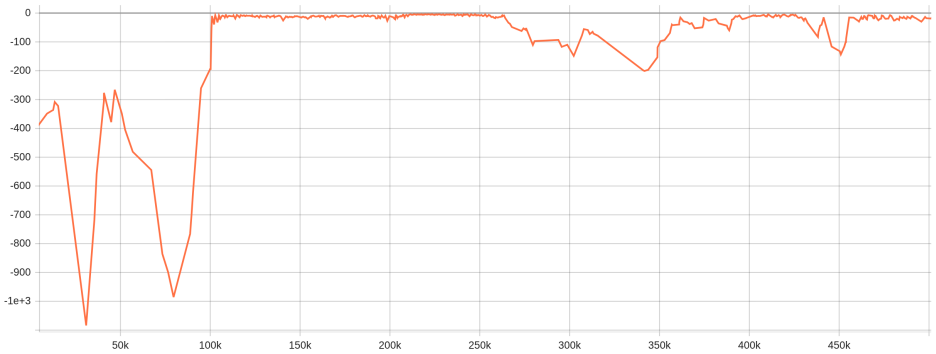

Figura 3 – Recompensa média dos episódios - SMC-S com ações incrementais

No que se refere ao corte de carga, apresentado em valores percentuais na Figura 4, observa-se uma forte oscilação no comportamento do agente, que não converge para uma estratégia consistente. Entre 100.000 e 250.000 passos, o agente apresenta uma atuação mais racional, reduzindo progressivamente o corte de carga até se aproximar de valores médios de corte próximos de 3% da demanda total. A partir desse ponto, entretanto, a política diverge, passando a propor cortes mais intensos e sem padrão definido. É válido destacar que o intervalo de menor corte médio, entre 250.000 e 400.000 passos, coincide com o intervalo de maior duração dos episódios, evidenciado na Figura 2. Esse resultado sugere que o ambiente induz o agente a associar os baixos índices de corte de carga a episódios mais longos, o que, em função do acúmulo de recompensas negativas, leva a uma deterioração da política aprendida. Cabe destacar, contudo, que manter o corte de carga nos níveis estritamente necessários é uma tarefa mais complexa, que exige um número maior de ações de controle por parte do agente.

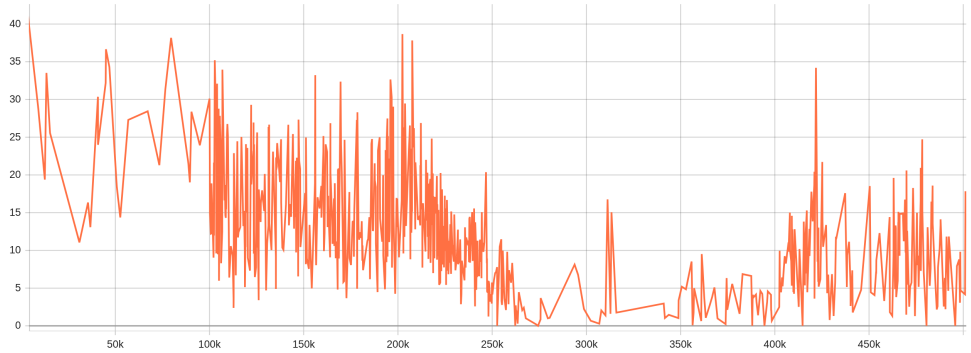

Figura 4 – Corte de carga percentual - SMC-S com ações incrementais

O agente apresentou bom desempenho nas demais métricas avaliadas, sendo capaz de manter a potência gerada na barra swing dentro dos limites físicos e de evitar sobrecargas na rede.

A instabilidade observada no treinamento levantou a hipótese de que a estratégia de aplicar ações de forma incremental e corrigir os valores aos limites físicos apenas no ambiente pode ter distorcido a percepção do agente sobre suas próprias decisões, já que, na versão SMC-S, ele não era

penalizado ao propor ações que violavam limites em barras PV. Outro fator que pode ter motivado a instabilidade é a definição do episódio em duas fases, estruturadas sobre uma linha temporal contínua, o que pode ter dificultado a aprendizagem ao reduzir a responsabilidade do agente pelo término dos episódios. Nessa configuração, é possível que o agente execute as ações necessárias para controlar um estado operativo, mas, ainda assim, receba uma recompensa insatisfatória em função da ocorrência de uma nova contingência ou da variação da curva de carga. Com base nesses hipósteses, foram testados ambientes progressivamente simplificados na tentativa de isolar possíveis falhas conceituais.

4.2 Ambiente de confiabilidade via SMC-S - Ações proporcionais

O treinamento do agente em um ambiente SMC-S com aplicação das ações de forma proporcional resultou em maior estabilidade na duração média dos episódios, apresentada na Figura 5. Nesse ambiente, o agente finaliza os episódios com uma média de cinco passos, aproximadamente a metade do observado quando as ações eram aplicadas de forma incremental. Esse resultado pode estar associado à maior amplitude das ações disponíveis, que permitem variações mais bruscas tanto na geração quanto na carga do sistema.

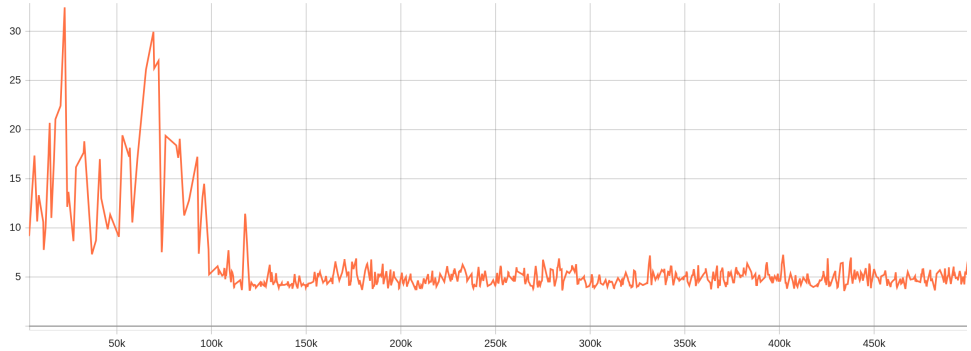

Figura 5 – Duração média dos episódios - SMC-S com ações proporcionais

A Figura 6 apresenta a evolução das recompensas médias por episódio. Os resultados demonstram que o desempenho do agente neste ambiente foi inferior ao do cenário anterior, resultado em uma recompensa média de aproximadamente -9. Embora o agente use menos passos por episódio, as recompensas piores indicam que ele adota soluções operativas mais custosas, distantes do estado ótimo do sistema. É verificada uma estabilidade nas recompensas médias logo após a fase de aquecimento, mas não há uma melhora progressiva desse valor ao longo das iterações, o que sugere que o agente pode ter ficado preso em um mínimo local. Neste ambiente, também foi testado um aumento no ruído de ação, mas os resultados permaneceram semelhantes.

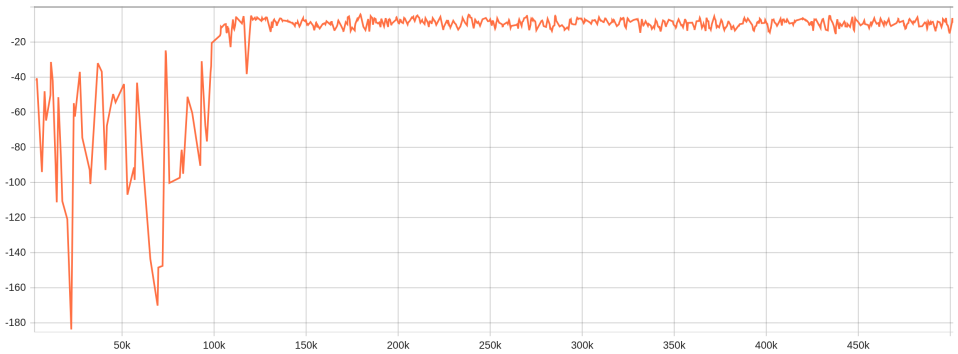

Figura 6 – Recompensa média dos episódios - SMC-S com ações proporcionais

A Figura 7 mostra a evolução do corte percentual de carga no sistema. Ao contrário do caso com ações incrementais, a curva exibe um comportamento desordenado, frequentemente ultrapassando 15% de corte. Embora se observe uma redução inicial, não há uma tendência clara de melhoria ao longo dos episódios. É válido ressaltar ainda que, em nenhum dos cenários avaliados, o corte de carga foi zerado. Isso indica que a política aprendida pelo agente também é ineficiente do ponto de vista de gerenciamento de cortes de carga.

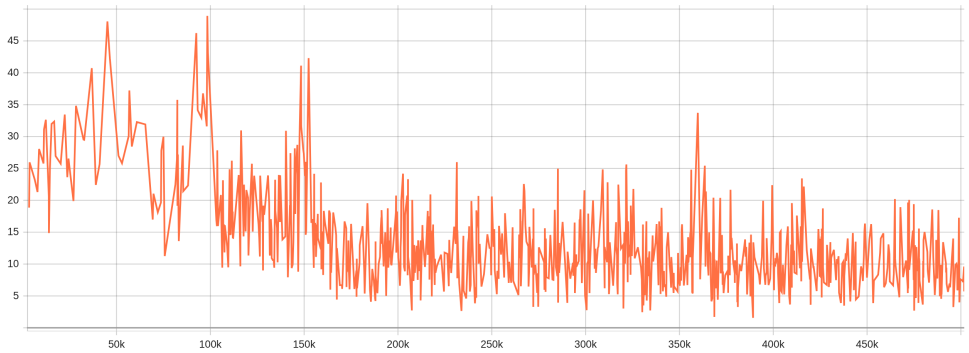

Figura 7 – Corte de carga percentual por episódio - SMC-S com ações proporcionais

Os resultados obtidos indicam que a estratégia de aplicar ações de forma proporcional não se mostrou eficaz para controlar o ambiente. É possível que a duração do treinamento, fixada em 500.000 passos, tenha sido insuficiente para que o agente aprendesse a utilizar esse mecanismo de controle.

4.3 Ambiente de confiabilidade via SMC-NS com ações incrementais

Com o objetivo de avaliar a influência da dependência temporal entre episódios no desempenho do agente, foi desenvolvido um ambiente de treinamento baseado em SMC-NS. Conforme discutido na Seção 2.1, nessa abordagem a amostragem de estados é realizada de forma aleatória e independente a cada episódio. Nesse ambiente, adotaram-se ações incrementais, tendo em vista o melhor desempenho observado na SMC-S. A expectativa era de que essa dinâmica simplificasse o treinamento do agente, ao delimitar de maneira mais clara os critérios de início e término dos episódios. No entanto, os testes mostraram uma dificuldade consideravelmente maior de aprendizagem.

Tanto a recompensa quanto a duração média dos episódios apresentaram alta variabilidade e nenhum indício de convergência, o que evidencia a inconsistência do processo de aprendizado.

A Figura 8 apresenta a evolução da duração média dos episódios. Observa-se que, logo após o início do treinamento, a duração dos episódios reduz de uma média de 50 para uma média de aproximadamente 35 passos. Contudo, esse valor apresenta forte oscilação ao longo do processo, sem tendência de estabilização.

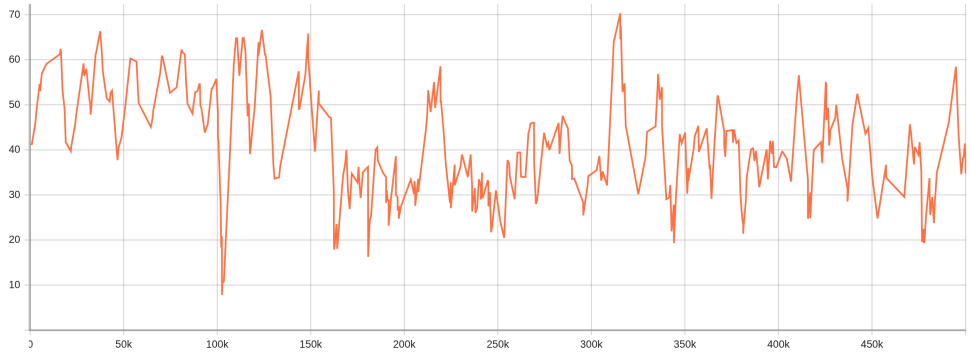

Figura 8 – Duração média dos episódios - SMC-NS com ações incrementais

Como consequência da maior duração dos episódios, as recompensas médias obtidas foram significativamente inferiores às observadas na SMC-S. Além disso, não se verificou qualquer tendência de estabilização ao longo das iterações, o que indica que o agente não conseguiu compreender a dinâmica do ambiente nem estabelecer uma política eficaz. O corte de carga acompanhou essa alta variabilidade, mantendo-se em torno de 15% da demanda total do sistema.

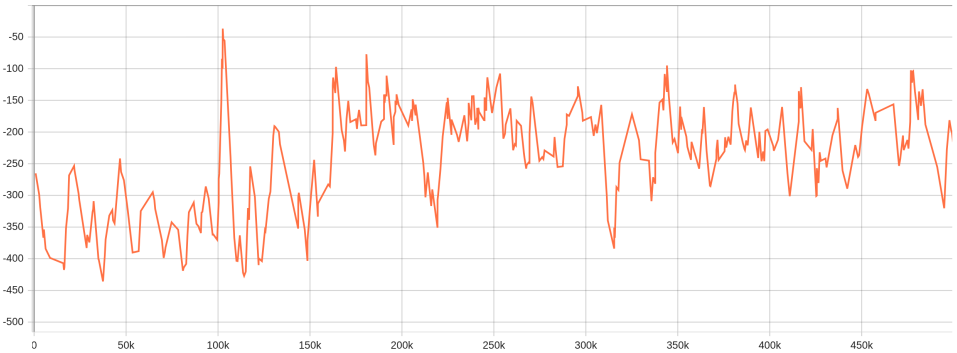

Figura 9 – Corte de carga percentual por episódio - SMC-NS com ações incrementais

O desempenho inferior nesse ambiente pode ser explicado pelo aumento da variância entre os estados. Como o *replay buffer* é preenchido com estados iniciais muito distintos, torna-se mais difícil para o agente identificar quais ações tomadas no período de aquecimento levaram a bons resultados, o que também dificulta a generalização do conhecimento para novos estados. Outro fator relevante é a ausência de referências na literatura para a escolha de hiperparâmetros no caso SMC-NS. Foram mantidos os mesmos parâmetros do ambiente sequencial, o que pode não ser adequado diante dos desafios próprios dessa abordagem.

5 CONCLUSÕES

Os experimentos realizados evidenciaram as dificuldades em aplicar diretamente algoritmos de aprendizagem por reforço ao problema de avaliação de confiabilidade composta em SEPs. Nos cenários testados com ambiente de simulação sequencial, observou-se que o agente apresentou capacidade limitada de aprendizado, tanto com aplicação incremental quanto proporcional das ações. Em alguns momentos, foi possível identificar indícios de comportamento racional, como a redução temporária do corte de carga ou a estabilização da duração dos episódios, mas esses avanços não se sustentaram ao longo do treinamento ou estagnaram em ótimos locais.

A tentativa de simplificação por meio da formulação não sequencial (MCS-NS) também não trouxe ganhos expressivos. Apesar de remover a dependência temporal, o agente continuou incapaz de desenvolver estratégias de controle eficientes devido à alta variabilidade dos cenários analisados. Além disso, a alta dimensionalidade do espaço de estados e ações se mostrou um impeditivo importante. Esse fator não apenas dificultou o aprendizado do agente, mas também comprometeu a própria análise dos resultados. O ideal seria treinar por um número muito maior de passos para avaliar com mais clareza o comportamento da política aprendida, mas as limitações computacionais impostas pelo tempo de GPU inviabilizaram essa abordagem.

Esses resultados sugerem que, embora a formulação proposta seja conceitualmente próxima da operação real do sistema, ela impõe barreiras ao aprendizado. A estagnação em ótimos locais, a complexidade de definir uma transição adequada para os episódios e a alta dimensionalidade do vetor de observação são fatores que dificultaram o processo de convergência. Esses resultados indicam a necessidade de explorar alternativas metodológicas, seja pela redução da dimensionalidade do problema ou por meio do desenvolvimento de funções de recompensa mais bem definidas, que forneçam incentivos mais claros ao agente para acelerar a finalização dos episódios e reduzir o corte de carga, evitando a estagnação em políticas sub-ótimas.

Além disso, outras formulações podem ser exploradas para incorporar a AR ao processo de avaliação de confiabilidade de SEPs, como, por exemplo, o treinamento do agente para a definição do despacho ótimo de geradores. Essa aplicação poderia ser concilidada com o método clássico de avaliação de forma a reduzir o número de execuções do FPO. Outra alternativa é realizar um pré-treinamento do agente com aprendizagem supervisionada, de forma a garantir maior estabilidade na fase inicial do aprendizado. É válido destacar, contudo, que a implementação dessas abordagens ainda dependeria da alta capacidade de processamento proporcionada pelas GPUs para viabilizar o treinamento dos modelos.

AGRADECIMENTOS

Ao concluir este trabalho, dedico esta seção para expressar minha sincera gratidão a todos que contribuíram para a sua realização.

Agradeço, em primeiro lugar, ao meu orientador, professor Dr. Fernando Aparecido de Assis, pela orientação, apoio e confiança durante o desenvolvimento deste trabalho.

Estendo meus agradecimentos aos colegas do Laboratório de Pesquisa em Energia Elétrica (LAENE), pela amizade, pelas discussões e por toda a ajuda que me deram ao longo do processo.

Dedico também um agradecimento aos meus pais, Inês e José, que tantas vezes me apoiaram e me incentivaram a seguir o chamado da engenharia, mesmo quando precisei abrir mão de grandes oportunidades para isso. Sou profundamente grato pela confiança que sempre depositaram em mim.

Agradeço, por fim, à Universidade Federal de São João del-Rei (UFSJ), pela estrutura e pelo ambiente acadêmico que possibilitaram a realização deste trabalho, e ao CNPq, pelo apoio financeiro concedido por meio da bolsa de iniciação científica.

REFERÊNCIAS

[1] Operador Nacional do Sistema Elétrico (ONS), *Procedimentos de Rede do Operador Nacional do Sistema Elétrico – ONS*, Jan. 2021. Versão 2020.12, aprovada pela REN ANEEL nº 903/2020, vigente a partir de 1º de janeiro de 2021.

[2] G. Li, Y. Huang, Z. Bie, and T. Ding, "Machine-learning-based reliability evaluation framework for power distribution networks," *IET Generation, Transmission Distribution*, vol. 14, no. 12, pp. 2282–2291, 2020.

[3] Fernando A. Assis, Alex J. C. Coelho, Lucas D. Rezende, Armando M. Leite da Silva, Leonidas C. Chaves, "Unsupervised machine learning techniques applied to composite reliability assessment of power systems," *International Transactions on Electrical Energy Systems*, 2021.

[4] R. Solheim, B. A. Høverstad, and M. Korpås, "Deep reinforcement learning applied to monte carlo power system reliability analysis," in *2023 IEEE Belgrade PowerTech*, pp. 1–6, 2023.

[5] R. Billinton and W. Li, *Reliability Assessment of Electric Power Systems Using Monte Carlo Methods*. Springer, 1994.

[6] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 2nd ed., 2018.

[7] R. Bellman, *Dynamic Programming*. Princeton University Press, 1957.

[8] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[9] G. A. Rummery and M. Niranjan, "On-line q-learning using connectionist systems," Tech. Rep. CUED/F-INFENG/TR 166, University of Cambridge, 1994.

[10] V. Mnih, K. Kavukcuoglu, D. Silver, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.

[11] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 1582–1591, 2018.

[12] R. T. S. T. F. of the Application of Probability Methods Subcommittee, "Ieee reliability test system," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-98, no. 6, pp. 2047–2054, 1979.