STU
SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA

Journal of Electrical Engineering, Vol. 77, No. 1, 2026, pp. 55-67
https://reference-global.com/journal/JEE

Formal validation for microgrid reliability based on TLA+

Muhammad Nasar, János Csatár

Ensuring the reliability of modern microgrids requires verification beyond conventional simulations. This paper introduces the novel application of TLA+, a formal specification language to model and verify a cyber-physical microgrid with six DERs and four priority loads. Our approach formally defines critical safety invariants such as Margin of Safety (MoS) ≥ 0 and automatically uncovers subtle design flaws, including MoS violations undetected by conventional testing. Experimental verification on a 25 kV system shows TLA+ can explore 1.79 million states in under 3 minutes, mathematically proving correctness under normal (MoS = +28) and attack scenarios (MoS = –2). State-space exploration ranges efficiently from minimal configurations (3,332 states, 8 s) to full-load conditions (1.79 M states, 171 s) with predictable growth indicated by anomaly percentages (0.19 %-65 %). Although validated on a single topology, the method establishes a provably correct foundation for microgrid design, advancing the shift from test-and-fix to correct-by-construction methodologies in resilient smart grid engineering.

Keywords: smart grids design, microgrid reliability, formal validation, TLA+ model checking

1 Introduction

Microgrids represent a transformative shift in energy infrastructure toward decentralized, greener, and more resilient power systems [1]. Their configurations are complex, comprising the main utility with diverse distributed energy resources (DERs) like Solar Photovoltaic (PV), Combined Heat and Power (CHP), and Battery Energy Storage Systems (BESS) [2]. These systems must meet strict reliability standards through seamless coordination of diverse, scalable DERs. IEC 61850 supports this coordination, but it heightens cyber-attack risks that impacts microgrid security and reliability [3-5].

Microgrid reliability goes beyond uptime. It requires stable power through resilient control and seamless mode transitions [6]. While grid-connected mode relies on the bulk system for voltage and frequency, islanded mode demands internal regulation using DERs [2]. Limited storage, controller precision, and sync speed [7], [8] make islanding a complex challenge and often needs custom solutions [9, 10]. Coordinated DER control is therefore essential.

Traditional methods like empirical testing and simulation offer limited assurance, as they only examine a finite set of scenarios and lack formal guarantees [11]. This is critical for complex systems like microgrids, where rare edge cases can cause cascading failures. Yet, simulation models often suffer from round-off errors and numerical instability, which distort results [10, 12, 13]. These shortcomings underscore the need for formal methods that enable rigorous, exhaustive validation of critical system properties.

In this regard, formal methods that use mathematical logic and a structured state-space approach to thoroughly verify systems offer strong assurance for safety-critical applications [14]. Unlike empirical methods that rely on observation or testing, formal methods prove that a system meets its specifications under all possible conditions [15]. However, their use in microgrids, particularly TLA+ for [i]validating dynamic behaviours like mode transitions, remains little. Most existing work depends on simulations, which cannot fully explore all system states or ensure robustness. This reveals a gap in applying TLA+ to model and verify microgrid operations under scenarios like islanding, DER integration, and cyber threats.

This paper advances microgrid reliability analysis by (i) surveying formal validation methods, and (ii) demonstrating TLA+ with TLC to verify stability in a medium-voltage microgrid. Comparisons with UPPAAL and SPIN highlight trade-offs in formalism choice. The key novelty and contributions lie in applying TLA+ to an underexplored area in formal methods, detailed below.

Department of Electric Power Engineering, Faculty of Electrical Engineering and Informatics,
Budapest University of Technology and Economics, Budapest, Hungary
Email: muhammad.nasar@edu.bme.hu

https://doi.org/10.2478/jee-2026-0006, Print (till 2015) ISSN 1335-3632, On-line ISSN 1339-309X

- Comprehensive synthesis of microgrid-oriented formal verification studies, mapping them by modeling paradigm, property class, and scalability limits.
- Definition and formalization of reliability validation criteria by specifying quantitative (supply-demand margin and state reachability), and qualitative (safety invariant and temporal stability) metrics within TLA framework to verify how reliability is verified.
- First complete TLA+ specification of a six-resource, four-load microgrid, expressed through explicit temporal properties and safety invariants.
- Guidelines for engineers on encoding DER controllers, network contingencies, and cyber events in TLA+, with reusable specification patterns.

The paper is structured as follows. Section 2 reviews related work on microgrid reliability and formal verification. Section 3 describes the case study, the TLA+ framework, specifications, and key properties. Section 4 presents verification results, scalability analysis, and deployment considerations. Section 5 concludes and suggests future research directions.

2 Related works

Recent literature emphasizes advanced computational techniques for optimizing microgrid control, energy management, and DER integration [2, 5, 7, 16]. However, rigorous formal methods for system-level reliability validation remain underutilized. While tools like NuSMV, PRISM, and Coq serve various verification needs, this section highlights TLA+, UPPAAL, and SPIN for addressing the discrete logic, timing, and concurrency challenges critical to microgrid fault tolerance, availability, and cyber resilience.

2.1 UPPAAL: Real-time and hybrid systems

UPPAAL models, simulates, and verifies systems as timed automata, with built-in clocks and deadlines ideal for time-critical microgrid tasks [17], where millisecond delays can impact stability [18]. It supports hybrid models combining continuous dynamics with discrete control, capturing behaviors like voltage and frequency shifts with switch operations. While it cannot fully verify nonlinear dynamics, simplified models still yield reliable safety insights. Its extension, UPPAAL SMC, uses statistical model checking to estimate property satisfaction through randomized simulations – well-suited for handling uncertainties from renewables, load fluctuations, and cyber threats [19].

Recent studies highlight UPPAAL's advantages. Paper [3] modeled a microgrid energy controller with unknown load bursts, revealing unsafe battery states missed by conventional tests. In [20], UPPAAL verified real-time IEC 61850 message flows and synthesized reliable relay trip schedules. Research [10] combined HOL4 theorem proving with UPPAAL event trees to assess cascading failures across large IEEE benchmarks. Study [21] modeled a substation-SCADA system and integrated a neural network (NN) anomaly detector; the formal model achieved 99% detection fidelity, compared to 91% for the NN alone – demonstrating the value of rigorous timing analysis in enhancing data-driven defenses. Together, these results place UPPAAL as a leading option for microgrid and broader power-system verification.

2.2 SPIN: Concurrent system verification

SPIN (Simple Promela Interpreter) is a widely used model checker for verifying concurrent and distributed systems, particularly communication protocols [22, 23]. Using the Promela language, it models complex process interactions like message passing and resource sharing. SPIN employs Linear Temporal Logic (LTL) to specify system properties, which it verifies using Buchi automata [24-26]. Its strength lies in detecting concurrency issues – such as deadlocks, race conditions, and synchronization errors – makes it well-suited for safety-critical systems where reliable inter-process communication is essential [27].

In power systems, researchers [28] used SPIN to assess fault tolerance in an IEEE 14-bus wide-area backup protection system, which demonstrates its effectiveness in managing complex smart grid coordination. SPIN's intuitive Promela language, state space reduction techniques, and detailed counter-examples make it valuable for debugging and refinement. However, it struggles with data-intensive models (e.g., discretizing battery SoC into ranges causes state space explosion and reduced accuracy) [29]. More critically, SPIN lacks native support for real-time behavior and continuous dynamics, which limits its suitability for cyber-physical systems like microgrids. Unlike UPPAAL, SPIN assumes correctness is unaffected by timing delays or execution speed, makes it less appropriate for verifying time-sensitive properties [30].

2.3 TLA+: Principles, strengths, and applications

TLA+ (Temporal Logic of Actions) works on set theory and temporal logic and models systems as state machines defined by logical actions. Its strength lies in specifying "what a system should do", not "how it does it" [31], which enables clear, high-level property definitions. This abstraction makes TLA+ well-suited for verifying logical correctness in complex, distributed systems like microgrids, where rigid timing constraints

(as in UPPAAL) can limit flexibility. Unlike SPIN, which emphasizes process interactions and ignores timing, TLA+ focuses on state transitions and system logic without relying on real-time semantics.

TLA+ effectively detects subtle design flaws missed by testing and is used by firms like Amazon and Datadog to verify complex systems [31, 32]. It supports both safety (e.g., "energy stays within limits") and liveness properties ("energy requests are eventually fulfilled"). Its model checker, TLC, exhaustively explores all possible states and generates counter-examples for detected violations. TLA+ also offers a proof system (TLA-PS) for formal correctness, though full liveness proof support remains under development [33].

In power systems, TLA+ has begun to gain interest, such as in verifying transactive energy coordination frameworks that enable DERs to negotiate energy exchange with neighbors[12]. While empirical and automata-based methods dominate, TLA+ offers a promising alternative with its ability to abstract complex behavior, handle distributed coordination and avoid rigid timing constraints, well-suited for validating scalable, fault-tolerant microgrids [34]. This research extends its use by applying TLA+ to implementation-level validation of microgrid reliability under fault and availability conditions.

Finally, TLA+ stands out for modeling complex systems with discrete and continuous behaviors without real-time constraints [33]. In contrast, UPPAAL handles real-time systems well [35] but scales poorly with many DERs, while SPIN excels at concurrency but assumes timing irrelevance and lacks support for continuous-time dynamics vital to microgrid stability [30].

2.4 Comparative summary and suitability

Literature consistently asserts that the effectiveness of formal methods is determined by the complexity of the system, the verification objective, and the level of abstraction of correctness. Although TLA+, UPPAAL, and SPIN are all categorized as model checkers, each operates on different semantics: temporal logic action, timed automata, and process-based concurrency, respectively. Because of these funda-mental differences, head-to-head quantitative bench-marks are not applicable to realistic microgrid models. Metrics such as the number of states or verification time reflect more on the coding details and reduction techniques of each tool than on differences in the correctness being tested.

Consequently, quantitative comparison is meaningful only in a restricted sense, limited to tool-centric measures (e.g., memory and processor con-sumption, exploration time under fixed bounds) and not to semantic measures of microgrid reliability. In the context of the microgrid model studied in this paper, global safety and liveness properties, timing feasibility, and concurrency behavior cannot be quantitatively compared across those tools, as they are expressed and interpreted differently in each formalism. Instead, these tools must be compared qualitatively in terms of suitability, while quantitative metrics are interpreted as diagnostic indicators of tool behavior, not as measures of verification strength.

This distinction is summarized in Tab. 1, which explicitly separates what aspects of microgrid verification can be meaningfully compared from those that cannot. As shown, TLA+ excels at abstract reasoning about system-wide safety and liveness without embedding implementation-level timing assumptions. UPPAAL is most effective for timing-critical and hybrid behaviors, where continuous clocks and stochastic extensions are required, albeit with reduced modeling flexibility. SPIN is particularly well suited for concurrency and protocol-level correctness but becomes less effective for hybrid, data-rich microgrid models.

Table 1. Comparative analysis of formal methods for microgrid reliability validation

Category	TLA+ (TLC) [15], [33], [34]	UPPAAL (SMC) [36], [37], [38], [39], [40]	SPIN [25], [26], [29], [41]
Formalism	Temporal logic of actions, set theory	Timed automata	Promela + LTL
Best suited for	High-level distributed systems	Real-time and hybrid systems	Concurrent protocols
Core strengths	Safety & liveness invariants, fairness, abstraction	Continuous time, hybrid dynamics, stochastic checking	Concurrency, interleavings, LTL checking
Main limitations	No native real time; state explosion	Scalability limits; rigid models	No native real time; limited data abstraction
Time & stochasticity	Discrete/abstract time; no native stochasticity	Continuous clocks; native SMC	Discrete events; limited stochasticity
Quantitative comparability	Tool-level metrics only (e.g., time, memory under bounds)	Same (not semantic strength)	Same (not semantic strength)
Typical microgrid role	Coordination logic & global reliability	Timing feasibility & dynamics	Protocol and interaction logic

These observations reinforce a key insight: Microgrid reliability depends on many interdependent aspects (logic, coordination, timing, physical dynamics, and uncertainty) so no single formal method can cover them all simultaneously. Therefore, the choice of tool should be driven by verification objectives, not bench-mark numbers. In this context, TLA+ was chosen because its high abstraction and mathematical semantics allow for comprehensive reasoning about the safety and sustainability of distributed energy resources, something that tools focused on timing or communication cannot directly achieve.

2.5 Model description

We study a 25 kV microgrid under cyberattack, connected to a 120 kV sub-transmission via a 15 MVA transformer [8]. It includes six DERs (CHP, PVs, BESS) and serves four loads – L1, L2 (critical, 3.2 MW each) and L3, L4 (non-critical, 3.2 MW and 4 MW), as shown in Fig. 1. Assuming of 0.8 power factor, they become 4, 4, 4, and 5 MVA respectively.

The system uses a data acquisition unit and a rule-based controller to manage DERs in islanded mode but lacks relay protection and reconnection. It communicates via IEC 61850 GOOSE, exposing signals to cyberattacks that can disrupt microgrid operation.

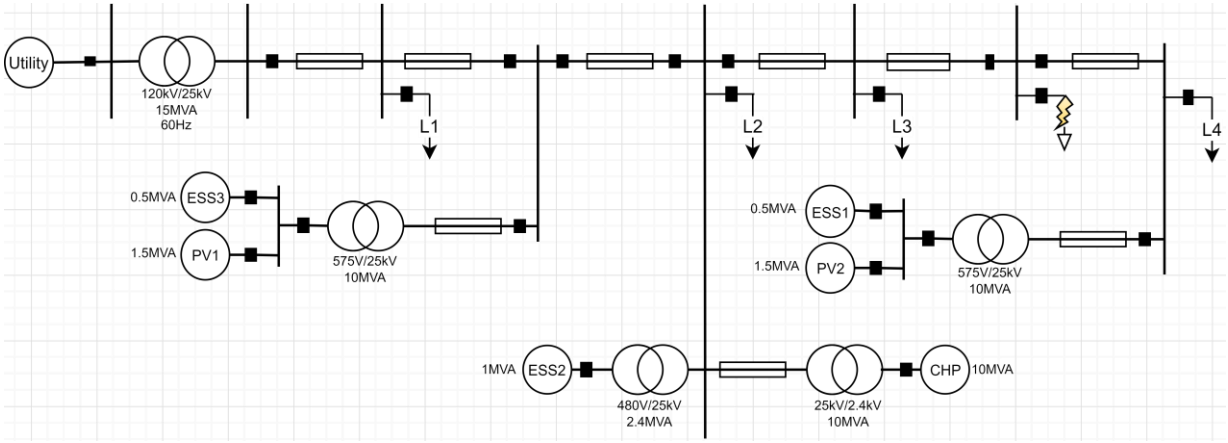

Fig. 1. The topology of the 25 kV microgrid studied

3 Model formalism

We defined discrete operational states based on interactions among key components (utility, PVs, BESS, CHP, and prioritized loads) in both grid-connected and islanded modes. Using defined max/min values, we generated input combinations and outputs. Table 2 shows selected islanded states (1-7, 15, 16). Values align with Fig. 1.

The Margin of Safety (MoS) is the power balance of supply and demand: $MoS = TotS - TotD$ (in MVA). A valid MoS falls within $0 < MoS \leq TotS$, indicating normal operation. Next, we define each component's operating states.

Table 2. State operations in islanded mode

State	Supply (MVA)					Demand (MVA)					MoS
	PV12	CHP	ESS12	ESS3	TotS	L1	L2	L3	L4	TotD	
1	3	10	1	1	15	4	4	4	0	12	3
2	3	10	0	1	14	4	4	4	0	12	2
3	3	10	0	0	13	4	4	4	0	12	1
4	0	10	0	0	10	4	4	0	0	8	2
5	3	0	0	0	3	0	0	0	0	0	3
6	3	0	–1	–1	1	0	0	0	0	0	1
7	0	0	1	1	2	0	0	0	0	0	2
15	0	0	0	0	0	0	0	0	0	0	0
16	1.5	10	0.5	1	15	4	8	4	0	16	–1

3.1 Photovoltaic (PV)

PV generates electricity based on solar radiation and temperature, with output rising under strong sunlight and dropping in cloudy or nighttime conditions. Its operation typically falls into three stages, depending on load demand and available supply.

$$PV\ State = \begin{cases} PV_{off} & if\ PV_{gen} = 0 \\ PV_{min} & if\ (PV_{gen} > 0) \wedge (PV_{gen} \leq (Loads + BESS)) \\ PV_{max} & if\ (PV_{gen} > 0) \wedge (PV_{gen} > (Loads + BESS)) \end{cases} \quad (1)$$

Here *PVoff* (no output), *PVmin* (partial output), and *PVmax* (surplus). In TLA+, only *PVoff* and *PVmax* are used, as *PVmin* falls within the range of these two states.

3.2 Battery energy storage system (BESS)

The BESS regulates energy fluctuations [8], with its state of charge (SoC_{BESS}) indicating available energy as a percentage of capacity [3]. When SoC_{BESS} is within SoC_{min} and SoC_{max} and TotS exceeds demand, the BESS charges; otherwise, it discharges to support supply. The controller prevents overcharge and deep discharge to protect the battery. BESS has three operating states: charging, discharging, and idle or standby. Its transitions between these operating states is determined in Eqn. (2).

$$State = \begin{cases} Charging & if\ (SoC_{min} < SoC_{BESS} < (SoC_{max}) \wedge (Dem < Sup) \\ Idle\ if\ ((SoC_{BESS} \leq SoC_{min}) \wedge (Dem \geq Sup)) \vee ((SoC_{BESS} \geq SoC_{max})) & \\ \quad \wedge (Dem < Sup) & \\ Discharging & if\ (SoC_{min} \leq SoC_{BESS} \leq SoC_{max}) \wedge (Dem \geq Sup) \end{cases} \quad ..(2)$$

Here, SoC_{min} and SoC_{max} define the allowable charge range (e.g., 25-95%), SoC_{BESS} is the current charge level, Dem refers to loads (L1-L4), and Sup includes PV, CHP, and BESS. As idle lies between them, only charging and discharging states are modelled.

3.3 Combined heat and power (CHP)

CHP systems are fuel-based, dispatchable units that generate electricity and reuse waste heat. They offer reliable and controllable output, ideal for supporting microgrids in islanded mode. To formalize CHP in TLA+, we assume constant fuel availability, reducing its operation to two states: CHP_{ON} and CHP_{OFF}. The CHP turns ON when total demand is non-zero and PVs + BESSs supply is insufficient (i.e., PVs+BESSs < TotD ∧ TotD ≠ 0).

$$CHP = \begin{cases} ON & if\ (PV_{gen} + SoC_{BESS} < TotD) \wedge (TotD \neq 0) \\ OFF & if\ (PV_{gen} + SoC_{BESS} \geq TotD) \vee (TotD = 0) \end{cases} \quad (3)$$

Here, CHP is the CHP state (ON/OFF), PV_{gen} is the total PV generation, and *TotD* is the total microgrid demand.

3.4 Load characteristic

This study considers four loads (L1-L4) with ratings of 4 MVA, 4 MVA, 4 MVA, and 5 MVA, respectively (Fig. 1). Load priority ranges from critical (L1, L2) to non-critical (L3, L4), as summarized in Tab. 3. As *TotS* represents the total available supply, then the demand can be expressed in Eqn. (4).

Table 3. Demand components and their priority

Load	Priority	Operating State	Use Case
L1	1st	4 MVA	Hospitals
L2	2nd	4 MVA	Factories
L3	3rd	4 MVA	Housing
L4	4th	5 MVA	Housing

$$TotD = \begin{cases} 0 & if\ TotS = 0, \\ L_1 & if\ TotS \geq L_1 \wedge TotS < (L_1 + L_2), \\ L_1 + L_2 & if\ TotS \geq (L_1 + L_2) \wedge TotS < (L_1 + L_2 + L_3), \\ L_1 + L_2 + L_3 & if\ TotS \geq (L_1 + L_2 + L_3) \wedge TotS < (L_1 + L_2 + L_3 + L_4), \\ L_1 + L_2 + L_3 + L_4 & if\ TotS \geq (L_1 + L_2 + L_3 + L_4). \end{cases} \quad (4)$$

Here, TotD represents active loads based on supply: 0 (none), L1 (only highest priority), L1+L2 (first two), L1+L2+L3 (first three), or L1+L2+L3+L4 (all) as supply suffices.

3.5 Abnormal states and invariant

State 16 in Tab. 2 denotes abnormal operation where *TotD* ≥ *TotS* caused by human errors or attacks related to false data injection (FDI). In FDI, reported *TotD* exceeds the real demand ($TotD_{real}$), which leads the controller to incorrect decisions despite sufficient TotS. Any attack types may also act as FDI if they result invalid measurement.

In TLA+, invariants ensure system safety, while temporal properties track behavior over time. Defined from supply-demand constraints (PV, BESS, CHP, L1-L4), these properties validate microgrid reliability. *MoS* = 0 signals an attack, and the model ensures *MoS* stays ≥ 0, triggering shutdowns if violated. *MoS* < 1 flags potential threats, guiding state-space validation, defined in Tabs. 4 and 5.

Table 4. Invariant properties

Invariant	Constraint	Formal State
Supply-demand balance	Demand ≤ Supply, and Demand ≥ 0	$TotD \leq TotS \wedge TotD \geq 0$
CHP activation rule	CHP turns ON if PV + BESS can't meet demand	$(TotS_{PV+BESS} < TotD) \wedge TotD \neq 0$
Battery safety	Battery charge stays between min/max limits	$SoC_{min} < SoC_{BESS} < SoC_{max}$
PV generation limits	PV only produces power during daytime/sunny	$(PV_{gen} \geq 0) \wedge (PV_{gen} \leq PV_{max})$
Load priority	Critical loads (L1, L2) get power first	L1 > L2 > L3 > L4
Attack detection	Fake demand > real demand → anomaly	$TotD_{fake} > TotD_{real}$

In this case, $TotD_{fake}=k\times TotD_{real}$ ($k>1$), where $TotD_{fake}$ is the fake demand injected by an attacker, k is the multiplier, and $TotD_{real}$ is the actual demand. By comparing $TotS$ to $TotD_{real}$ regarding the load priority, the system response can be formulated in Eqn. (5).

$$TotD_{fake} = \begin{cases} 0 & if\ TotS = 0, \\ L_1 & if\ TotS \geq L_1 \wedge TotS < (L_1 + L_2)/k, \\ L_1 + L_2 & if\ TotS \geq (L_1 + L_2)/k \wedge TotS < (L_1 + L_2 + L_3)/k, \\ L_1 + L_2 + L_3 & if\ TotS \geq (L_1 + L_2 + L_3)/k \wedge TotS < TotD_{fake}, \\ L_1 + L_2 + L_3 + L_4 & if\ TotS \geq TotD_{fake}. \end{cases} \quad (5)$$

When the state is normal, the controller compares TotS with TotDreal by load priority; under attack ($k > 1$), it acts on inflated $TotD$, altering decisions as thresholds are divided by k (e.g., $TotS$ = (L1+L2)/2 if k=2).

Table 5. Temporal properties

Temporal	Constraint	Formal state
PV liveness	As long as the sun shines, the PV will generate power.	$Sunlight > 0 \rightarrow PV_{gen} > 0$
BESS liveness	BESS charges when SoC is below maximum and supply exceeds demand.	$TotS > TotD \wedge SoC_{BESS} < SoC_{max} \rightarrow BESS_{charge} > 0$
CHP liveness	If more than other supplies are needed, CHP will turn on.	$TotS_{PV+BESS} < TotD \wedge TotD \neq 0 \rightarrow CHP_{ON}$
Load liveness	If the supply is sufficient again, all priority loads will be supplied.	$TotS \geq TotD \rightarrow TotD_{active} = TotD$
Attacked load recovery	If the attack stops, the load will be active again.	$TotD_{fake} = TotD_{real} \rightarrow L_{active}$

3.6 TLA+ specification

Our TLA+ model simulates supplying prioritized loads from multiple sources, using semaphores for synchronized, fair access. It ensures valid states, balances supply and demand, and halts when complete or insufficient. Four input parameters guide the process:

- PwrSupps: A sequence ⟨*p, q, r, s*⟩ of supply values, where each *p, q, r, s* ∈ R and ≥ 1.
- PwrValues: A sequence of sets {*0, a*} for each supply, with $a \in R$ and $a \geq 1$, representing possible contributions to total supply.
- Loads: The number of load demands.
- LoadValues: A set of possible load values based on the number of loads.

Figure 2 presents the TLA+ model: (a) input and invariants, (b) temporal properties. It uses nine inputs for power, supply, demand, and completion. Five invariants ensure consistency and capacity limits. Supply and demand run as fair, independent processes, followed by distribution that computes and validates MoS within safe bounds.

VARIABLES $Pwrs$, $PwrSupps$, $Supply$, $SupplyCnt$, Ls, $Demand$, MoS, $Finished$, pc

define statement

$InvariantCheck \triangleq Sum(SupplyCnt) = Sum(PwrSupps) \land Ls = Loads \land Sum(Supply) > 0$
$\land Finished \Rightarrow (MoS > 0 \land MoS \leq Sum(Supply))$
$UtilityCheck \triangleq Supply[1] \geq 0 \land Supply[1] \leq Utils * Max(UtilValues)$
$PVCheck \triangleq Supply[2] \geq 0 \land Supply[2] \leq PVs * Max(PVValues)$
$CHPCheck \triangleq Supply[3] \geq 0 \land Supply[3] \leq CHPs * Max(CHPValues)$
$ESSCheck \triangleq Supply[4] \geq 0 \land Supply[4] \leq ESSs * Max(ESSValues)$

(a)

```
fair process util ∈ 1 .. Utils
begin acompute:
    with uval ∈ UtilValues do
        Supply[1] := Supply[1] + uval ;
        SupplyCnt[1] := SupplyCnt[1] + 1 ;
    end with ;
end process ;

fair process pv ∈ (Utils + 1) .. (Utils + PVs)
begin bcompute:
    with pval ∈ PVValues do
        Supply[2] := Supply[2] + pval ;
        SupplyCnt[2] := SupplyCnt[2] + 1 ;
    end with ;

end process ;

fair process chp ∈ (Utils + PVs + 1) .. (Utils + PVs + CHPs)
begin ccompute:
    with cval ∈ CHPValues do
        Supply[3] := Supply[3] + cvalyCnt[3] := SupplyCnt[3] + 1 ;
    end with ;
end process ;

fair process ess ∈ (Utils + PVs + CHPs + 1) .. (Utils + PVs + CHPs + ESSs)
begin dcompute:
    with eval ∈ ESSValues do
        Supply[4] := Supply[4] + eval ;
        SupplyCnt[4] := SupplyCnt[4] + 1 ;
    end with ;
end process ;
```

```
fair process usages = 200
variables
    j = 0 ;
begin
    ucompute:  while j < Loads do
        with u ∈ LoadValues do
            Demand := Append(Demand, u) ;
        end with ;
        j := j + 1 ;
    end while ;
    Ls := Ls + Loads ;
end process ;

fair process distribute = 300
variables
    OriDemand = ⟨⟩ ;
    d = 0 ;
    Print_MoS = 0 ;
begin
    chk:
        await Sum(SupplyCnt) = Sum(PwrSupps) ∧ Ls = Loads ;
        OriDemand := Demand ;
        MoS := Sum(Supply) ;

    compute:
        while Finished = FALSE ∧ Sum(Demand) > 0 do
            d := Head(Demand) ;
            if (MoS − d) > 0 then
            MoS := MoS − d ;

            Demand := Tail(Demand) ;
            else
                Print_MoS := MoS − d ;
                print ⟨"PwrSupps:", PwrSupps, "[Utility,PV,CHP,ESS] => Max:",
                ⟨Max(UtilValues), Max(PVValues), Max(CHPValues), Max(ESSValues)⟩, "Supply:",
                Supply, "TotSupply:", Sum(Supply), ",Demands:", OriDemand, ",TotDemand:",
                Sum(OriDemand), ",Unprocessed:", Demand, ",MoS:", Print_MoS⟩ ;
                Finished := TRUE ;
            end if ;
        end while ;
end process ;

end algorithm
```

(b)

Fig. 2. TLA+ code segments defining (a) the input variables and invariant properties, (b) temporal properties

3.7 TLA+ workflow

The diagram in Fig. 3 illustrates that in the TLA+ framework, the model is treated as a discrete state transition system that verifies the reliability of the hybrid microgrid through invariants and temporal logic. Continuous energy balancing is reduced to a state machine with the tuple ⟨*Supply, Demand, MoS*⟩, while transition rules capture the physical and operational limits of the assets. The CONSTANTS module instantiates resource types – Utility, PV, CHP, ESS – with deterministic or stochastic distributions for stress testing. The PROCESSES module coordinates multi-source supply (P1-P4), while the COMPUTATION module models demand dynamics: CP1 manages the load queue and CP2 enforces the *MoS* - *d* > 0 invariant, which ensures security only when the supply margin remains above the incremental demand.

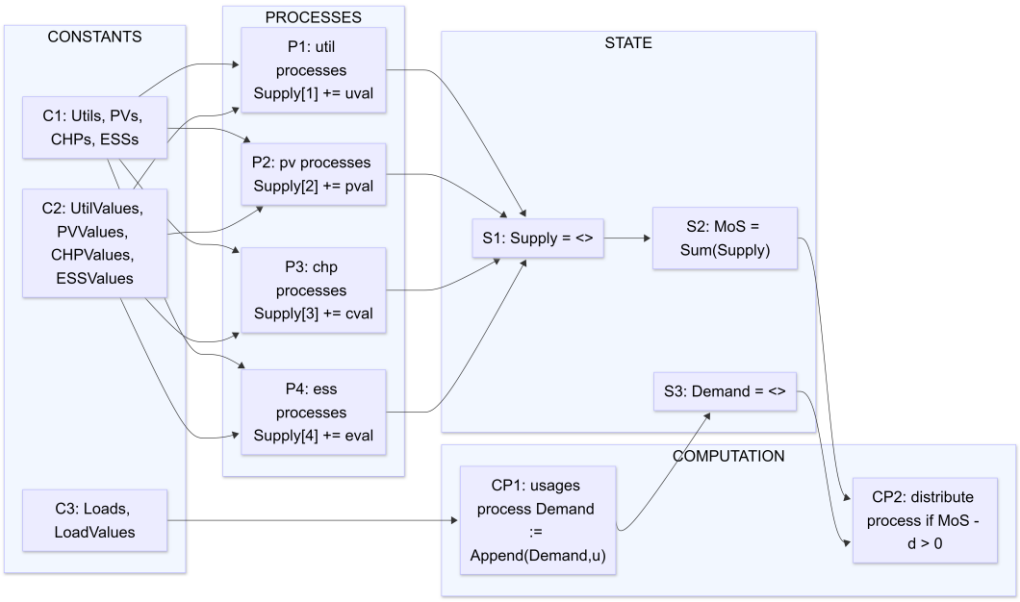

Fig. 3. TLA+ modeling workflow diagram for microgrids energy distribution system

The TLA+ verification flow assesses the reliability of the microgrid by traversing the entire state space to prove $MoS \geq 0$ invariance, ensuring that the supply margin is never negative under all fluctuation patterns. The separation of STATE variables (S1-S3) from PROCESSES enables stability analysis, demonstrating that reliability is determined not only by aggregate capacity but also by the coordination of the CP2 protocol. TLC checks confirm the absence of deadlocks or resource exhaustion, and guarantee that all demand is eventually satisfied when the stochastic supply satisfies the ergodic condition of the load profile. Thus, empirical metrics such as load loss probability are mapped to formal invariants that prove that the cyber-physical control logic is capable of maintaining energy balance under modeled disturbances.

4 Experiment

We conducted six scenarios to evaluate system performance under varying inputs and state space progressions.

4.1 Experiment scenarios and result

This study tests two scenarios: Grid-Connected Mode, where the utility supplies stable power, and Islanded Mode, where internal sources (PV, CHP, ESS) provide power. Source and load variations simulate real-world conditions as outlined in Section 3.

The system simulates power supply under varying loads by tracking state counts, unique states, and state space diameter (longest shortest path). Large diameters may challenge model checkers and need careful analysis. Scenarios, defined by input parameters (supply types, power, loads), guide exploration of state spaces with supplies and loads drawn from predefined ranges. State counts and queue sizes are monitored at each diameter stage.

Tables 6 and 7 summarize microgrid reliability testing via TLA+, covering operation, supply, demand, and results. *TotS* and *TotD* represent total supply and demand, with *MoS* (*TotS* – *TotD*) typically positive. Values are doubled to avoid fractions. Scenarios include normal (within limits) and abnormal (beyond limits), which triggers shutdown. N1 models a full shutdown with all components off ($MoS = 0$), exploring 3,332 states in 8 s. N2 represents stable grid-connected operation ($TotS = 60$, $TotD = 32$, $MoS = 28$), with 1,791,140 states explored in 2 m 51 s, indicating high interaction complexity. A1 uses only Utility (30) with all loads on ($TotD = 32$, $MoS = -2$), leading to quick shutdown after 42,081 states in 16 s. A2, with Utility off and other supplies totalling 30, also yields $MoS = -2$ but explores 1,166,530 states in 1m 52s, showing greater complexity before failure. In N3, with A2's supply and L4 off, demand drops to 24 ($MoS = 6$), yielding 568,891 states in 57 s. N4, using A1's supply and also deactivating L4, achieves the same *MoS* with fewer resources, exploring just 20,489 states in 5 s – making it the most efficient case.

Table 6. Experiment scenarios

Operation	Supply								Demand					MoS
	UTL	PV1	PV2	CHP	ESS1	ESS2	ESS3	TotS	L1	L2	L3	L4	TotD	
N1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N2	30	3	3	20	1	2	1	60	8	8	8	8	32	28
A1	30	0	0	0	0	0	0	30	8	8	8	8	32	–2
A2	0	3	3	20	1	2	1	30	8	8	8	8	32	–2
N3	0	3	3	20	1	2	1	30	8	8	8	0	24	6
N4	30	0	0	0	0	0	0	30	8	8	8	0	24	6

N1=Normal1, N2=Normal2, A1=Abnormal1, A2=Abnormal2, N3=Normal3, N4=Normal4

Table 7. Experiment results

Operation	States explored	Time (s)	Diameter	States/ s	Mem (GB)	CPU (%)
N1	3,332	8	15	416	0.8	68
N2	1,791,140	171	19	10,474	12.7	73
A1	42,081	16	19	2,630	4.2	73
A2	1,166,530	112	19	10,415	11.3	72
N3	568,891	57	17	9,980	7.1	73
N4	20,489	5	17	4,098	2.4	72

4.2 Validation analysis

In normal operations, the system performs best in N2 with full supply and *MoS* of 28. N3 and N4 maintain stability by reducing demand despite partial supply loss. In contrast, A1 and A2 show system failure with MoS of –2 due to insufficient supply, highlighting the importance of redundancy for reliability. The TLA+ results show that complexity rises with more active components (e.g., N2, A2), yet the model effectively identifies failures (A1/2) and confirms stability in reduced-demand cases (N3/4). This highlights the importance of supply redundancy, demand flexibility, and TLA+'s effectiveness in verifying microgrid reliability.

Figure 4 shows the microgrid's behavior across modes. N1, with no active components, explores just 3,332 states (0.19% of N2) in 8 seconds, with $MoS = 0$. In contrast, N2, the baseline for full operation, explores 1,791,140 states in 171 seconds with $MoS = 28$, which reflects a stable, complex system. A1 and A2 expose failure conditions with negative *MoS* (–2), indicating unmet demand. A1 shows a low-complexity failure (2.35% anomalies), while A2 reveals a complex grid-connected failure (65.13% anomalies). In contrast, N3 and N4 maintain a positive MoS by reducing demand, with lower anomaly rates (31.76% and 1.14%). These emphasize the need for supply-demand balance and demonstrate TLA+'s value in identifying both failures and optimal states.

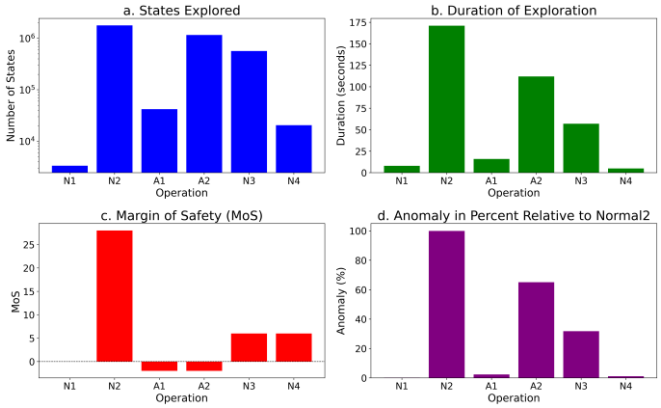

Fig. 4. State accumulation trend visualizing (a) states found, (b) exploration time, (c) margin of safety, (d) percent anomaly relative to N2 (Normal2)

The validation shows grid loss increases complexity by 2.67%, while demand management reduces it. It highlights the importance of grid support and load prioritization for reliability and risk mitigation.

4.3 Scalability analysis

We evaluate the scalability of the TLA+/PlusCal framework through analysis of the execution platform, TLC results, and specification structure. Experiments on Windows 10 with TLC v2.19 on an i7 system with 16 GB of RAM and 256GB SSD show that the primary limitation comes from explicit state storage, not CPU throughput. In all scenarios, CPU usage stabilizes at 68-73%, while memory usage increases with the number of states, confirming that the verification is memory-bound.

TLC maintains a stable throughput of around 10^4 states/second on large executions (>10^6 states). On small configurations, the throughput drops to 400-4,000 states/second due to fixed initialization costs, not algorithmic inefficiencies. With N_s as the number of states and M as the peak memory, the largest normal case (N_s≈1.79×10^6, M≈12.5 GB) yields the following Estimated memory footprint per state:

$$\bar{m}_{state} \approx \frac{M}{N_s} \approx 7\ \text{kB/state} \tag{6}$$

Given the practical stability threshold of TLC at approximately 85-90% of available RAM, the safe upper bound on this hardware is therefore:

$$N_s^{max} \approx \frac{0.85 \times 16\ \text{GB}}{7\ \text{kB}} \approx 2 \times 10^6\ \text{states} \tag{7}$$

This bound closely matches the observed termination envelope, confirming that scalability is dominated by state retention rather than transition evaluation. The limitation stems not from the model checker, but from the semantic structure of the PlusCal specification. The model instantiates multiple concurrent processes that are fair for utility, PV, CHP, and ESS, each selecting values non-deterministically from a limited domain. Load demand is also modeled as a sequence, not an aggregate scalar. Therefore, the growth of its state space follows:

$$|S| \approx (\prod_{i\in(U,PV,CHP,ESS)} . |values_i|^{count_i}) . |LoadValues|^{Loads} \tag{8}$$

which is exponential in both the number of DERs and the number of loads.

From a scalability perspective, each additional DER introduces an approximate multiplicative factor of $|Values_i|$, and each additional load introduces a factor of $|LoadValues|$. Formally, the growth factors are:

$$\frac{|S_{k+1}^{DER}|}{|S_k^{DER}|} \approx |Values_i|, \quad \frac{|S_{l+1}^{Load}|}{|S_l^{Load}|} \approx |LoadValues| \tag{9}$$

Finally, projection in Tab. 8 shows that scalability increases almost linearly with RAM capacity, while CPU capacity yields diminishing benefits. This encourages the use of platforms with large memory or semantic abstractions such as aggregation and symmetry reduction for larger microgrid configurations.

Table 8. Scalability projection by hardware class

Hardware configuration	TLC state limit	Notes
16 GB RAM (current)	~2 M	Memory-bound
32 GB RAM	~4-5 M	Near-linear scaling
64 GB RAM	~9-10 M	CPU begins to dominate
SSD	No major impact	TLC is RAM-resident

4.4 Validation against operational expectation

We qualitatively validated the TLA+/PlusCal model by linking the TLC exploration results to patterns commonly observed in microgrid simulations and operations. Across six scenarios, TLC explored 3.3×10^3 to 1.79×10^6 states, using up to 12.7 GB of memory, and reaching nearly 10^4 states/second in the largest scenario.

In normal scenarios with high variability (N2, N3), exploration of 0.5-1.8 million states showed the system remained capable of meeting supply while generating numerous interleaves between source and demand, consistent with stable but rich time-domain simulations due to variability in renewable energy and storage operation. In contrast, the constrained scenarios (N1, N4) generated only 10^3-10^4 states and completed within seconds, which reflects an evolution space narrowed by operational limits.

The abnormal scenarios were also consistent with stress behavior. Abnormal1 terminates after 4.2×10^4 states when the *MoS* is depleted, resembling a rapidly reaching-limit overload simulation. Abnormal2 explores over 1.1 million states with high memory usage, reflecting broader and more complex stress conditions under fluctuating generation. In both scenarios, the TLC traces show the *MoS* decreasing monotonically toward depletion, consistent with the reserve collapse pattern in the simulation.

Across all scenarios, the safety invariants (source output limits, aggregate supply consistency, and *MoS* limits) remain satisfied, even in explorations beyond 10^6 states. This aligns with the power and reserve balance checks in the operational study, which confirms that the model eliminates physically unrealistic behavior.

Overall, the state space variation, resource usage, and shutdown patterns are consistent with the distinction between nominal and stressed regimes in the microgrid simulation. This alignment reinforces the realism of the abstract model and demonstrates that the verified properties capture relevant operational behavior. This supports the use of TLA+ as a design-stage analysis tool for microgrids.

4.5 Study limitations

This study has several limitations that open up avenues for further research. First, although tested on a realistic 25 kV microgrid with heterogeneous DERs and prioritized loads, the model is still limited to a single architectural configuration; its scalability to larger systems with mesh topologies, more nodes, or hierarchical control has not been validated.

Second, the specification uses deterministic and discrete abstractions and therefore does not capture the stochastic uncertainty and dynamics of real-world hybrid microgrids – including renewable energy variability, load fluctuations, and communication and control latency. Higher-fidelity analysis requires probabilistic and continuous-time modelling.

Third, the performance evaluation was conducted solely on TLC and does not allow for a fair quantitative comparison with UPPAAL or SPIN. Due to differences in semantics and modeling styles, cross-tool results remain qualitative; semantically aligned benchmarks are needed for a truly equitable evaluation.

Finally, the model is not yet coupled with power system simulation or hardware-in-the-loop (HIL) platforms – a recent common simulation tool in power systems, which limits validation against physical dynamics. Therefore, further work needs to focus on more scalable models, semantically aligned cross-tool benchmarking, and tighter integration with HIL simulation.

5 Conclusion

This study demonstrates that a scalability-aware, invariant-based formal workflow can transform formal verification from merely descriptive analysis into a practical design methodology for microgrid controllers. Using the PlusCal/TLA+ unified model and explicit mapping between verification costs, model structure, and hardware limits, we demonstrate that global properties (such as supply-demand balance and safety margins) can be verified in a large operating space, aligning with patterns observed in real microgrid simulations.

The main contribution lies in the methodological approach: TLA+ is positioned as an early design tool, where invariants shape the architecture and scalability analysis guides abstraction. The results confirm that verification success is determined more by state design and invariant formulation than by raw tool capabilities. We also show that cross-tool comparisons (TLA+, UPPAAL, SPIN) require semantic alignment, making raw quantitative benchmarks inadequate.

From an industry perspective, this approach enables early assurance regarding safety margins, resource utilization, and robustness to demand variability, which are critical factors for certification and risk reduction. Overall, formal verification combined with scalability analysis and operational validation emerges as a viable engineering tool for next-generation microgrids, with further work directed at semantically aligned cross-tool benchmarking and tighter integration with power simulation and hardware-in-the-loop platforms.

Acknowledgement

We sincerely thank I Made Putrama (Ganesha University of Education) for his pivotal contribution to the development and testing of the TLA+ specifications according to the proposed design, which significantly enhanced the rigor and quality of this work.

References

[1] M. Sandelic, S. Peyghami, A. Sangwongwanich, and F. Blaabjerg, “Reliability aspects in microgrid design and planning: Status and power electronics-induced challenges,” *Renewable and Sustainable Energy Reviews*, vol. 159, p. 112127, May 2022, doi: 10.1016/j.rser.2022.112127.

[2] R. Asri, H. Aki, and D. Kodaira, “Optimal operation of shared energy storage on islanded microgrid for remote communities,” *Sustainable Energy, Grids and Networks*, vol. 35, p. 101104, Sep. 2023, doi: 10.1016/j.segan.2023.101104.

[3] K. Fellah and R. Abbou, “Modelling and Formal Verification Approach for Microgrid Energy Management Systems under Random Load,” *IFAC-PapersOnLine*, vol. 58, no. 1, pp. 270–275, 2024, doi: 10.1016/j.ifacol.2024.07.046.

[4] H. Hayati, A. Ahadi, and S. M. Miryousefi Aval, “New concept and procedure for reliability assessment of an IEC 61850 based substation and distribution automation considering secondary device faults,” *Front. Energy*, vol. 9, no. 4, pp. 387–398, Dec. 2015, doi: 10.1007/s11708-015-0382-6.

[5] S. M. H. Rostami, M. Pourgholi, and H. Asharioun, “Enhancing resilience of distributed DC microgrids against cyber attacks using a transformer-based Kalman filter estimator,” *Sci Rep*, vol. 15, no. 1, p. 6815, Feb. 2025, doi: 10.1038/s41598-025-90959-4.

[6] T. Sun, J. Lu, Z. Li, D. L. Lubkeman, and N. Lu, “Modeling Combined Heat and Power Systems for Microgrid Applications,” *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 4172–4180, Sep. 2018, doi: 10.1109/TSG.2017.2652723.

[7] Sk. A. Shezan *et al.*, “Optimization and control of solar-wind islanded hybrid microgrid by using heuristic and deterministic optimization algorithms and fuzzy logic controller,” *Energy Reports*, vol. 10, pp. 3272–3288, Nov. 2023, doi: 10.1016/j.egyr.2023.10.016.

[8] L. Zhang *et al.*, "Cybersecurity Study of Power System Utilizing Advanced CPS Simulation Tools".

[9] Y. Wang, C. Deng, Y. Liu, and Z. Wei, "A cyber-resilient control approach for islanded microgrids under hybrid attacks," *International Journal of Electrical Power & Energy Systems*, vol. 147, p. 108889, May 2023, doi: 10.1016/j.ijepes.2022.108889.

[10] M. Abdelghany and S. Tahar, "Reliability Analysis of Smart Grids Using Formal Methods," in *Handbook of Smart Energy Systems*, M. Fathi, E. Zio, and P. M. Pardalos, Eds., Cham: Springer International Publishing, 2023, pp. 147–163. doi: 10.1007/978-3-030-97940-9_81.

[11] M. Aryanfar, M. Fotuhi-Firuzabad, S. Azad, P. Dehghanian, and S. Peyghami, "Resilience vs. reliability metrics in power systems," in *Future Modern Distribution Networks Resilience*, Elsevier, 2024, pp. 67–90. doi: 10.1016/B978-0-443-16086-8.00004-X.

[12] A. Ransil, M. Hammersley, and F. M. O'Sullivan, "Improving system resilience through formal verification of transactive energy controls," in *2020 IEEE PES Transactive Energy Systems Conference (TESC)*, Portland, OR, USA: IEEE, Dec. 2020, pp. 1–5. doi: 10.1109/TESC50295.2020.9656940.

[13] A. Mahmood, O. Hasan, H. R. Gillani, Y. Saleem, and S. R. Hasan, "Formal reliability analysis of protective systems in smart grids," in *2016 IEEE Region 10 Symposium (TENSYMP)*, Bali, Indonesia: IEEE, May 2016, pp. 198–202. doi: 10.1109/TENCONSpring.2016.7519404.

[14] M. Gleirscher, J. Van De Pol, and J. Woodcock, "A manifesto for applicable formal methods," *Softw Syst Model*, vol. 22, no. 6, pp. 1737–1749, Dec. 2023, doi: 10.1007/s10270-023-01124-2.

[15] R. Bögli, L. Lerena, C. Tsigkanos, and T. Kehrer, "A Systematic Literature Review on a Decade of Industrial TLA+ Practice," in *Integrated Formal Methods*, vol. 15234, N. Kosmatov and L. Kovács, Eds., in Lecture Notes in Computer Science, vol. 15234. Cham: Springer Nature Switzerland, 2025, pp. 24–34. doi: 10.1007/978-3-031-76554-4_2.

[16] A. S. Satapathy *et al.*, "Emerging technologies, opportunities and challenges for microgrid stability and control," *Energy Reports*, vol. 11, pp. 3562–3580, Jun. 2024, doi: 10.1016/j.egyr.2024.03.026.

[17] F. Shokri-Manninen, L. Tsiopoulos, J. Vain, and M. Waldén, "Integration of iUML-B and UPPAAL Timed Automata for Development of Real-Time Systems with Concurrent Processes," in *Rigorous State-Based Methods*, vol. 12071, A. Raschke, D. Méry, and F. Houdek, Eds., in Lecture Notes in Computer Science, vol. 12071, Cham: Springer International Publishing, 2020, pp. 186–202. doi: 10.1007/978-3-030-48077-6_13.

[18] G. Sugumar, R. Selvamuthukumaran, M. Novak, and T. Dragicevic, "Supervisory Energy-Management Systems for Microgrids: Modeling and Formal Verification," *EEE Ind. Electron. Mag.*, vol. 13, no. 1, pp. 26–37, Mar. 2019, doi: 10.1109/MIE.2019.2893768.

[19] A. Hachiro, M. Nakamura, K. Sakakibara, T. Matumoto, and R. mosTakano, "Modeling and Verification of Variable Multi-Lane Intersection Controls for Autonomous Vehicles Using Uppaal SMC," in *2024 International Conference on Machine Learning and Cybernetics (ICMLC)*, Miyazaki, Japan: IEEE, Sep. 2024, pp. 291–296. doi: 10.1109/ICMLC63072.2024.10935218.

[20] G. Kunz, J. Machado, E. Perondi, and V. Vyatkin, "A Formal Methodology for Accomplishing IEC 61850 Real-Time Communication Requirements," *IEEE Trans. Ind. Electron.*, vol. 64, no. 8, pp. 6582–6590, Aug. 2017, doi: 10.1109/TIE.2017.2682042.

[21] P. Kreimel, O. Eigner, F. Mercaldo, A. Santone, and P. Tavolato, "Anomaly detection in substation networks," *Journal of Information Security and Applications*, vol. 54, p. 102527, Oct. 2020, doi: 10.1016/j.jisa.2020.102527.

[22] Q. Wang, "Applying SPIN checker on 5G EAP-TLS authentication protocol analysis," *ComSIS*, vol. 21, no. 1, pp. 21–36, 2024, doi: 10.2298/CSIS230611068W.

[23] M. Rashid, M. Qadeer, H. Raza, I. Shabbir, I. Rasool, and N. A. Zafar, "Formal Modeling and Verification of IoT-based Smart Transport System using SPIN Model Checker," in *2024 IEEE 1st Karachi Section Humanitarian Technology Conference (KHI-HTC)*, Tandojam, Pakistan: IEEE, Jan. 2024, pp. 1–6. doi: 10.1109/KHI-HTC60760.2024.10481884.

[24] N. O. Garanina *et al.*, "A Temporal Logic for Programmable Logic Controllers," *Aut. Control Comp. Sci.*, vol. 55, no. 7, pp. 763–775, Dec. 2021, doi: 10.3103/S0146411621070038.

[25] L. Huang, Y. Liang, H. Huang, Q. Wang, S. Qian, and R. Zhao, "Application of Formal Method in Grid Cyber Physical Systems," in *2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, Chongqing, China: IEEE, Dec. 2020, pp. 1075–1080. doi: 10.1109/ITAIC49862.2020.9338880.

[26] B. Vlaovič and A. Vreže, "Discrete Time Model for Process Meta Language with Fictitious-Clock," *Applied Sciences*, vol. 12, no. 6, p. 2990, Mar. 2022, doi: 10.3390/app12062990.

[27] J. Esparza, J. Křetínský, J.-F. Raskin, and S. Sickert, "From linear temporal logic and limit-deterministic Büchi automata to deterministic parity automata," *Int J Softw Tools Technol Transfer*, vol. 24, no. 4, pp. 635–659, Aug. 2022, doi: 10.1007/s10009-022-00663-1.

[28] S. T. Hamman, K. M. Hopkinson, and J. E. Fadul, "A Model Checking Approach to Testing the Reliability of Smart Grid Protection Systems," *IEEE Trans. Power Delivery*, pp. 1–1, 2016, doi: 10.1109/TPWRD.2016.2635480.

[29] T. Do, A. C. M. Fong, and R. Pears, "How Effective Is Model Checking in Practice?," in *Proceedings of the 6th International Conference on Evaluation of Novel Approaches to Software Engineering*, Beijing, China: SCITEPRESS - Science and Technology Publications, 2011, pp. 239–244. doi: 10.5220/0003467402390244.

[30] G. J. Holzmann, *The spin model checker: primer and reference manual*, 4th print. Boston, Mass. Munich: Addison-Wesley, 2008.

[31] R. Otoni, I. Konnov, J. Kukovec, P. Eugster, and N. Sharygina, "Symbolic Model Checking for TLA+ Made Faster," in *Tools and Algorithms for the Construction and Analysis of Systems*, vol. 13993, S. Sankaranarayanan and N. Sharygina, Eds., in Lecture Notes in Computer Science, vol. 13993. , Cham: Springer Nature Switzerland, 2023, pp. 126–144. doi: 10.1007/978-3-031-30823-9_7.

[32] C. Newcombe, T. Rath, F. Zhang, B. Munteanu, M. Brooker, and M. Deardeuff, "How Amazon web services uses formal methods," *Commun. ACM*, vol. 58, no. 4, pp. 66–73, Mar. 2015, doi: 10.1145/2699417.

[33] L. Lamport, *Specifying systems: the TLA+ language and tools for hardware and software engineers*. Boston: Addison-Wesley, 2003.

[34] S. Sajjad, N. Akhter, and L. Sajjad, "Formal Modelling and Model Checking of a Flood Monitoring and Rescue System: A Case Study of Safety-Critical System," *VFAST trans. softw. eng.*, vol. 12, no. 3, pp. 114–137, Sep. 2024, doi: 10.21015/vtse.v12i3.1871.

[35] G. Behrmann, A. David, and K. G. Larsen, "A Tutorial on Uppaal," in *Formal Methods for the Design of Real-Time Systems*, vol. 3185, M. Bernardo and F. Corradini, Eds., in Lecture Notes in Computer Science, vol. 3185. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 200–236. doi: 10.1007/978-3-540-30080-9_7.

[36] I. Grobelna, K. Gajewski, and A. Karatkevich, "A Systematic Review on the Applications of Uppaal," *Sensors*, vol. 25, no. 11, p. 3484, May 2025, doi: 10.3390/s25113484.

[37] J. Martins, A. Platzer, and J. Leite, "Statistical Model Checking for Distributed Probabilistic-Control Hybrid Automata with Smart Grid Applications," in *Formal Methods and Software Engineering*, vol. 6991, S. Qin and Z. Qiu, Eds., in Lecture Notes in Computer Science, vol. 6991. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 131–146. doi: 10.1007/978-3-642-24559-6_11.

[38] M. Althoff, "Benchmarks for the Formal Verification of Power Systems," presented at the Proceedings of 9th International Workshop on Applied Verification of Continuous and Hybrid Systems (ARCH22), pp. 26–7. doi: 10.29007/tg4s.

[39] S. Chakraborty, J.-P. Katoen, F. Sher, and M. Strelec, "Modelling and statistical model checking of a microgrid," *Int J Softw Tools Technol Transfer*, vol. 17, no. 4, pp. 537–554, Aug. 2015, doi: 10.1007/s10009-014-0345-y.

[40] M. Naeem, R. Gu, C. Seceleanu, K. Guldstrand Larsen, B. Nielsen, and M. Albano, "Energy-Efficient Motion Planning for Autonomous Vehicles Using Uppaal Stratego," in *Theoretical Aspects of Software Engineering*, vol. 14777, W.-N. Chin and Z. Xu, Eds., in Lecture Notes in Computer Science, vol. 14777, Cham: Springer Nature Switzerland, 2024, pp. 356–373. doi: 10.1007/978-3-031-64626-3_21.

[41] N. Suresh Kumar and G. Santhosh Kumar, "Abstracting IoT protocols using timed process algebra and SPIN model checker," *Cluster Comput*, vol. 26, no. 2, pp. 1611–1629, Apr. 2023, doi: 10.1007/s10586-022-03963-y.

Received 8 November 2025