

Formal Modeling with Verification of Smart Courtroom Building System (SMCBS) Using UML and TLA+ Specification

Shahbaz Ahmad¹, Nadeem Akhtar¹,
Adeel Ahmad^{1,*}, Saima Abdulla¹ and
Saman Iftikhar²

¹Department of Computer Science
& Information Technology, Faculty
of Computing, The Islamia
University Bahawalpur, Bahawalpur,
Punjab 63100, Pakistan

²Faculty of Computer Studies, Arab
Open University, Saudi Arabia

*E-mail: adeelmcs@gmail.com

Received for publication
July 04, 2024.

Abstract

The growing trend of interconnected subsystems has brought significant interest in designing smart systems with the ability to build fully controlled attention upon them. Designing and verifying a smart building system presents unique challenges, including ensuring security for sensitive areas, managing diverse user access levels, and integrating multiple subsystems while maintaining system integrity and performance. There is also a need to work and focus on the goal of a smart, safe, and secure courtroom environment for all stakeholders, particularly in developing countries, such as Pakistan. There are also problems of security, safety, and energy saving in the infrastructure of the smart building systems. This study presents a proposed model for a secure, safe, and energy-saving courtroom building, and the model utilizes Unified Modeling Language (UML) diagrams to visualize the system's components or objects and interaction sequences of actions. Additionally, formal methods are used, specifically Temporal Logic of Actions (TLA+), to write specifications as per the requirements of specifications and to rigorously verify the system's properties to ensure its correctness. In the context of the smart courtroom building, security measures, such as password-protected entry, user access level, and safety, are implemented for the courtroom building. The system also manages building-wide lighting and temperature control while incorporating a subsystem of smoke detection to enhance safety during a fire in the courtroom. The model's properties, such as correctness, are rigorously verified using the TLA+ toolbox and built-in Model Checker (TLC) after applying the specification as per the requirements of the infrastructure of a building. So, we have developed a novel integration of UML modeling and TLA+ specification for comprehensive system representation and verification. This is a robust set of formal specifications that address the unique security and functionality requirements of a courtroom environment. We have successfully verified the system properties using the TLC model checker, ensuring the correctness and safety of the proposed smart courtroom building system.

Keywords

formal modelling, TLA+, formal methods, UML, energy efficiency, verification

I. Introduction

The diverse usage of Internet of Things (IoT) stands as a cutting-edge technology with favorable applications across numerous industries. One particularly key sector is smart buildings within the territory of IoT, artificial intelligence (AI), and machine learning. Day-to-day, IoT finds applications in diverse industries, such as smart cities, smart homes, smart grids, e-health, quality management, and transportation. Using IoT in smart buildings emerges as a cost-effective and efficient solution. Key considerations for commercial spaces, such as comfort, accessibility, security, and energy management, align naturally with IoT-based technologies. This chapter provides an overview of IoT technology, its architecture, and the monitoring of smart buildings. The aim is to lay the groundwork for future studies that leverage spatial data collected from diverse IoT devices, including temperature sensors and security cameras, already operational in buildings. Effective moisture control is crucial for maintaining a healthy indoor environment in residential buildings [1–5]. Creating necessitates a smart system combining various subsystems to model a fully well-designed smart system. The process of verifying and validating such a smart system is a considerable challenge. Many researchers have researched various intelligent systems, including the automation of houses, offices, libraries, autonomous vehicles, campuses, and smart airport systems. However, many of these systems have not been researched with a focus on ensuring privacy and security. This study introduces a novel proposal for a smart courtroom building system (SMCBS), which not only incorporates the typical features of smart systems but also fulfills the safety properties, ensuring a private and secure environment, particularly in Pakistan. In this era of research, they engage the Unified Modeling Language (UML) along with formal methods to explain the objects, sequences of actions, and the behavior of the intelligent courtroom system [6]. UML serves as an informal, object-oriented modeling approach that incorporates various pictorial practices, such as state, activity, sequence, and entity diagrams. It contains a wide range of diagrams in both structured methods and the object-oriented domain. As a collection of graphical models, UML speaks to diverse properties of design, with the primary focus on two types of models: (i) behavioral and (ii) structural models. In this framework, they emphasize the behavior of the models, a state machine of automation in verification that effectively explains the active behavior of the system [7–9]. Numerous

methods exist for verification and validation, with the adoption of formal methods being particularly favored by many researchers. This preference is attributed to the effectiveness of formal methods in designing systems with minimal bugs. Over the last decade, the use of formal methods has gained traction and has been embraced by several reputable companies [10, 11]. Formal methods play a crucial role in supporting the verification, development, and specification tools of both software and hardware systems. These methods are grounded in mathematics and utilize software tools to demonstrate that the formal implementation model of a system accurately aligns with its specifications. The key objective in system development is to simplify and clarify complexities, and formal methods excel in achieving this goal. What sets formal methods apart is their incorporation of formal verification and the system's correctness before its acceptance. The trustworthiness of formal verification tools is notably high, making them viable alternatives to the existing verification framework [12].

Formal methods are extensively engaged to ensure the correctness, safety, and liveness of various systems, with many verification processes using formal methods. These methods excel at identifying errors and bugs in systems that may go undetected by other approaches. Well-known companies have relied on formal methods for an extended period to model, verify, and validate the complexities of their systems. The initial step involves the modeling of the system by translating it into an abstract mathematical model, specifying its properties and requirements to use a formal specification language. Our key motivation is to utilize formal methods not only to achieve the safety and security of our enhanced smart system [13–16].

Safety Critical Systems (SCSs) refer to systems whose failure could result in catastrophic environmental consequences, potentially leading to the loss of lives. In recent decades, the prevalence of SCSs has increased significantly across various domains, including nuclear reactors, the avionics industry, and the military sector. With advancing technology, there is a gradual reduction in manual labor due to increased automation. This shift toward automation underscores the need for heightened attention to safety and reliability by both researchers and industries. Formal methods serve as crucial techniques in the design and verification of SCSs. Unlike alternative system design approaches, formal methods provide a guarantee of system correctness through the application of mathematical proofs. As technology continues to evolve, the reliance on formal methods

becomes paramount in ensuring the robustness and reliability of SCSs [17].

In this era of research, they engage the UML along with formal methods to illustrate the objects, sequences of actions, and the behavior of the intelligent courtroom system [6]. UML serves as an informal, object-oriented modelling approach that incorporates various pictorial practices, such as state, activity, sequence, and entity diagrams. It contains various diagrams in both structured methods and the object-oriented domain. As a collection of graphical models, UML speaks to diverse properties of design, with the primary focus on two types of models: (i) behavioral and (ii) structural models. In this framework, the emphasis is on the behavior of the models, the state machine of automation in verification that effectively explains the active behavior of the system [7–9].

Creating necessitates a smart system combining various subsystems to model a fully well-designed smart system. The process of verifying and validating such a smart system is a considerable challenge. Numerous methods exist for verification and validation, with the adoption of formal methods being particularly favored by many researchers. This preference is attributed to the effectiveness of formal methods in designing systems with minimal bugs. Over the last decade, the use of formal methods has gained traction and has been embraced by several reputable companies [10, 11].

Formal methods play a crucial role in supporting the verification, development, and specification tools of both software and hardware systems. These methods are grounded in mathematics and utilize software tools to demonstrate that the formal implementation model of a system accurately aligns with its specifications. The key objective in system development is to simplify and clarify complexities, and formal methods excel in achieving this goal. What sets formal methods apart is their incorporation of formal verification and the system's correctness before its acceptance. The trustworthiness of formal verification tools is notably high, making them viable alternatives to its existing verification framework [12].

Formal methods are extensively engaged to ensure the correctness, safety, and liveness of various systems, with many verification processes using formal methods. These methods excel at identifying errors and bugs in systems that may go undetected by other approaches. Well-known companies have relied on formal methods for an extended period to model, verify, and validate the complexities of their

systems. The initial step involves the modeling of the system by translating it into an abstract mathematical model, specifying its properties and requirements to use a formal specification language. Our key motivation is to utilize formal methods not only to achieve the safety and security of our enhanced smart system [13–16].

SCSs refer to systems whose failure could result in catastrophic environmental consequences, potentially leading to the loss of lives. In recent decades, the prevalence of SCSs has increased significantly across various domains, including nuclear reactors, the avionics industry, and the military sector. With advancing technology, there is a gradual reduction in manual labor due to increased automation. This shift toward automation underscores the need for heightened attention to safety and reliability by both researchers and industries. Formal methods serve as crucial techniques in the design and verification of SCSs. Unlike alternative system design approaches, formal methods provide a guarantee of system correctness through the application of mathematical proofs. As technology continues to evolve, the reliance on formal methods becomes paramount in ensuring the robustness and reliability of SCSs [17].

Smart and energy-efficient buildings are increasingly being enhanced as the construction industry influences decarbonization, leading to an impressive interest in sophisticated control systems that balance energy use with occupant comfort. Although there are many developments in this field, there has not been a comprehensive review of modern control systems for these buildings, particularly regarding innovative energy systems, machine learning applications, and controls that focus attention on resident needs. This latest progress in smart building mechanisms covers everything from data collection through sensors to building automation and energy modeling. It looks closely at different machine-learning approaches and their practical applications. The findings also reflect how control systems can make buildings more reliable, secure, and adaptable to changing conditions, including their role in fault detection and climate transformation response. Particular awareness is presented to systems that rank occupant comfort and behavior. The review concludes by deliberating key challenges, such as discovering the right balance between system complexity and performance, the role of AI in adapting to climate change, and how occupant-focused controls can progress relief and energy efficiency. This research specifies a systematic overview of current developments in

smart building control systems, helping to guide the construction industry regarding a more sustainable future [18].

China is targeting carbon neutrality by 2060, enhanced by significant policy ideas announced at COP26, and the call among governments to improve energy efficiency in existing buildings has never been higher. There may be many studies about energy-efficient renovations, but there is still no comprehensive overview of how incentives are being applied in practice. Major gaps include the lack of analysis on how national and local policies are linked, insufficient scrutiny of recent policy developments and impacts, and little understanding of the multiple barriers to these renovations. While the complexities have made addressing these challenges difficult, this study utilizes comprehensive policy analysis to clarify how national and local incentives are connected (as they are seen in this study) as well as have shifted over time. It further examines pilot city-based initiatives and delivers best practice success stories and lessons learned as use cases. We next examined the challenges confronting governments, homeowners, companies, and academics regarding these issues and provided some recommendations. The purpose of the study is to contribute to informing future policy and regulation on energy-efficient retrofit business decisions in China [19].

Specific challenges in designing smart courtroom systems: smart courtroom systems face unique challenges, such as maintaining strict access control, ensuring data privacy, and integrating various subsystems, such as temperature control and smoke detection, while adhering to legal and security standards.

The limitations of existing approaches:

On the contrary, the previous studies have addressed smart building systems; they often lack the rigorous formal verification needed for critical environments, such as courtrooms, and do not adequately address the integration of multiple security-sensitive subsystems.

This study presents a novel approach that: Combines UML's visual modeling capabilities with Temporal Logic of Actions (TLA+)'s formal verification power, providing a comprehensive method for designing and validating complex smart systems. Develops a set of formal specifications tailored to the unique requirements of a courtroom environment, including strict access control and safety measures. Demonstrates the successful verification of critical system properties using the TLC model checker, ensuring the proposed system meets stringent

security and functionality requirements. Table 1 shows the summary of the literature review related to smart buildings.

a. Research contributions

Maintaining concepts, TLA+ treats properties and specifications alike; expressing both as logical formulas is a more powerful full specification language. Our proposed model engages TLA plus a formal specification language after using UML design to demonstrate the system's abstract model. The application of formal modeling and verification techniques to a SMCBS, which has not been previously explored in the literature. The integration of UML and TLA+ for modeling and verifying a complex, multicomponent smart building system. The TLA+ toolbox will be utilized to prove the correctness and verification of our proposed model. The SMCBS includes various subsystems. Each presiding officer, staff, advocate, accused, and litigant entrance requires authentication via a username and password, and visitors can only enter if authenticated by a reception employee for safety. Once the arrival of the first presiding officer or court employee, the light system activates automatically through sensors, and temperature sensors regulate the temperature of the building. Another subsystem of smoke detection ensures safety after detecting any smoke in case of a fire.

Organization: The rest of the study is structured as follows: Section II describes the related work in this field, Section III represents the UML-based smart courtroom building model, Section IV describes the formal specification using TLA+, Section V discusses the TLA+ model analysis, and finally Section VI contains the conclusion and future work.

II. Background of Studies

In this study, a famous researcher Leslie Lamport [33] has used the TLA+ specification language and refined modeling language to design for concurrent and distributed systems, enabling precise system descriptions through mathematical expressions. It proves instrumental in identifying crucial design errors that may be challenging to detect and costly to rectify at the code level. TLA+ was created by Lamport specifically for formal reasoning, describing the distributed algorithms. His work, *Specifying Systems* in 2002, provides insights into TLA+, its usage, and percentage money for the fulfillment of their desired needs and to enhance efficiency and boost employee

Table 1: Summary regarding the review of literature on smart buildings

Authors	Year	Usage of methodology	Main findings/viewpoints	Limitations
Rodríguez Gracia et al. [20]	Apr 2023	AI techniques in sustainable buildings	Explores how AI can enhance the sustainability of buildings through a comprehensive review of the existing literature. Utilized bibliometric analysis to identify key trends.	Relies on bibliometric data, which may not capture the practical, real-world effectiveness of AI solutions.
Khan et al. [21]	Dec 2022	Critical event library for building fire safety	Developed a framework for firefighting in buildings based on significant fire events. The framework aims to predict future incidents using a database of past events.	Based on historical data, which may not account for new building materials and designs or evolving safety standards.
Liu et al. [22]	Jan 2023	Data-driven smart photovoltaic systems in buildings	Summarized literature on data-driven approaches with integrated photovoltaic systems, covering data collection, analysis, prediction, and optimization.	The focus on data-driven methods does not address the practical implementation and cost considerations for large-scale use.
Xia et al. [23]	Sep 2022	GIS and BIM integration for smart cities	Reviewed data integration methods for combining GIS and BIM technologies in urban planning, emphasizing sustainable city design through digital twins.	The study is largely conceptual and does not provide practical case studies on the implementation of these technologies.
Pinto et al. [24]	Feb 2022	Transfer learning in smart building applications	Reviewed the use of transfer learning and deep learning in smart building technologies, categorizing existing research into four major application areas.	Primarily focuses on theoretical applications without a detailed discussion on the challenges of real-world deployment.
Dwivedi et al. [25]	Apr 2022	IoMT in healthcare buildings	Investigated the use of IoT in healthcare buildings, discussing potential benefits and challenges in creating smart healthcare environments.	Limited discussion on data security and privacy concerns associated with IoT in healthcare settings.
Huseien and Shah [26]	Jan 2022	5G technology for smart energy management	Discussed the potential of 5G technology to support energy management in smart buildings, with a focus on Singapore's governmental support for this technology.	Focused on the Singapore context, which may not be applicable to other regions with different regulatory environments.

(Continued)

Table 1: Continued

Authors	Year	Usage of methodology	Main findings/viewpoints	Limitations
Aguilar et al. [27]	Nov 2021	AI for energy management in buildings	Reviewed AI applications for self-managing energy systems in buildings, categorizing studies by task areas of monitoring and decision-making.	Limited to AI technologies; does not address the challenges of integrating these systems with existing infrastructure.
Malagnino et al. [28]	Aug 2021	IoT and BIM for sustainable building environments	Explored how building information modeling and the IoT can improve building sustainability, analyzing literature from 2015 to 2020.	Focused on older literature, potentially missing recent advancements and ongoing research in this rapidly evolving field.
Stopps et al. [29]	Jan 2021	Occupant-centric HVAC controls in residential buildings	Reviewed the research on smart HVAC systems that focus on occupant comfort, comparing experimental studies with existing commercial technologies.	Emphasizes research gaps without fully exploring how these systems can be effectively integrated into current homes.
Dakheel et al. [30]	Oct 2020	Smart building features and performance indicators	Reviewed the technologies and KPIs that define intelligent building features and discussed the challenges of retrofitting existing buildings.	The review is general; it lacks specific case studies or quantitative assessments of different smart building technologies.
Dong et al. [31]	Sep 2019	Smart building sensing systems	Reviewed how indoor environmental sensors impact energy efficiency and comfort, covering aspects regarding visual comfort and indoor air quality in the building.	The focus on sensors overlooks the challenges related to maintenance, data accuracy, and system integration.
Jia et al. [32]	May 2019	IoT adoption in smart buildings	Reviewed key projects and research on the changing and fixing of IoT technologies for smart buildings, also highlighting improvements in sites of academia and industry.	Lacks an evaluation for long-term sustainability and scalability of these IoT solutions in several different building types.
Our proposed work	Oct 2024	Adoption of IoT technologies, the formal model language of TLA+, specification with formal methods in smart courtroom building		

AI, artificial intelligence; BIM, Building Information Modeling; GIS, Geographic Information System; HVAC, Heating, Ventilation, and Air Conditioning; IoMT, Internet of Medical Things; IoT, Internet of Things; KPIs, key performance indicators; TLA+, Temporal Logic of Actions.

performance, certain crucial steps are implemented. The concept of smart offices involves modeling the office management system to gain a comprehensive understanding and visualization from various perspectives. UML diagrams are utilized to capture the operational and interactive aspects of a system, while Automata-based models automate system behavior in terms of states. Ensuring the smart office system's consistency, accuracy, and reliability is attained via formal methods. So, Vienna Development Method (VDM) and specification language play a key role in verifying and validating the proposed model, contributing to the overall effectiveness and dependability of the smart office system [34].

This study addresses all these challenges by focusing on developing a smart city sewerage system using IoT, UML, automata, and TLA+. The proposed sewerage system is constructed through "UML" activity diagrams, which are then transformed into Non-deterministic Finite Automata (NFA). Each junction is represented as a state, and water flow is depicted as a transition. A system state is considered safe if the wastewater flow between junctions follows a secure path, predicting water behavior. Near to formalizing the UML and automata-based models, we employ TLA+ with robust model-checking capabilities. The correctness of our proposed model is rigorously demonstrated through the use of TLC, a model-checking tool within the TLA+ toolbox, ensuring the effectiveness and reliability of the smart city sewerage system [35].

The recent advancements in technology have sparked considerable interest in the field of implementing intelligent systems. Researchers have explored various smart systems, including home, office, and parking areas, in response to this technological advancement. This system operates on a wireless fidelity (Wi-Fi) network utilizing the AllJoyn framework. It incorporates an asymmetric curve of cryptography for authentication to ensure secure operations. This authentication process facilitates user interaction, access, and control through an Android-based application program. Additionally, there is a proposal for a smart campus positioned around the mobile social network. This initiative involves integrating the community relations within the campus environment with an elastic system architecture design. The model considers the client side, representing mobile users, and the server side, representing both the social connections and services [36–39].

The decision-making process contains mixed-integer linear programming for every decision, resulting in reduced average parking costs and time spent

searching for a parking spot. Additionally, a sewage system was proposed using NFA, unified data modeling language, and TLA+. The system is designed to run a function as a subsystem within a broader range of the smart city framework. Furthermore, a survey is presented to cover the basic and foundational architecture, protocols, and technologies for an urban IoT, along with considerations on best-practice procedures and methodological resolutions for a project in the Padova Smart City in Italy [40, 41].

Several investigators have undertaken the modeling of intelligent courtroom buildings, primarily focusing on power consumption and management. Additionally, around the investigations into smart campus systems, emphasizing mobile learning, cloud-based learning, e-learning, and promoting the environmental awareness on campus [42–48].

The research work has proposed a system that varies slightly due to the distinctive environmental and the distinctive features compared to the elementary courtroom. The primary objective of this smart system is to integrate interconnected subsystems to create a comprehensive, smart, and autonomously controlled system. Safety stands out as a critical property that these systems must possess, particularly in the context of our SMCBS. Another essential characteristic to be incorporated into these smart systems is security. While many researchers tend to focus on achieving either safety or security, our proposed model aims to establish a system that is both secure and safe. Future discussions will explore further into the details of the proposed system. Various smart systems have undergone verification and validation through formal methods, utilizing different formal specification languages [49–51], such as Alloy, VDM, Z, and TLAplus. Latif et al. [52] applied TLA+ for modeling and verification of a smart parking system, while Rehman et al. [34] used the Vienna Development Method Specification Language (VDM-SL) toolbox for designing and verifying the smart office system [53].

In this study, they have applied the TLA+, which is a specification language for verification, and a choice was made for several reasons. First, when compared to other specification languages, TLA+ is noted for its simplicity and directness, requiring fewer layers of identification for modeling nested structures than, for instance, Alloy. While Alloy's expressiveness is somewhat limited in comparison, it compensates with the efficiency of the Alloy analyzer model checker, adept at handling extensive analyses that TLC may struggle with. Second, TLA+ allows editing details in specifications when needed and supports high-level functions. On the contrary, Alloy exhibits less flexibility and

lacks support for recursive functions. In comparison, Verifying Concurrent C (VCC) [54] enables the writing of “ghost code,” the C-programming language; however, it is considerably more talkative than TLA plus and Alloy. Another advantage of TLA+ is that it has a built-in a model checker, TLC, which determines the capability to handle extensive state spaces with acceptable throughput rates in testing. Unlike Event-B model checkers, VCC, B [55, 56], and TLC can efficiently operate multiple operations and are not restricted to single-memory processors. As the Alloy analyzer is faster and quicker than TLC, it encounters issues, such as hanging or crashing, in cases crucial for larger systems, presenting certain limitations.

The UML was first used in 1994 and afterward gained standardization by the Object Management Group (OMG) in 1997. It also achieved ISO standard status in 2005, and the current version is called UML-2. This provides a notation for structuring software design and evaluating existing designs. It establishes a common language among software professionals, facilitating detailed communication. UML diagrams offer a means to communicate software structure without exploring into code, aiding in onboarding new developers and interactions with management. In the current advancing digital age, information systems play an essential role in improving efficiency across various sectors, particularly within public facilities. So, the UML approach serves as an efficient guide to designing systems for attaining increased effectiveness. This approach explores ascertaining issues, user requirements, and both functional and non-functional phases of the system. This denotes a set of graphical notations to use for describing and designing the software through the use of diagrams. It is particularly appropriate for object-oriented software, and its applicability extends to different types of software. It is also used to develop a system in software engineering and is a pictorial language to explain and document a system. The requirements in states are specific, and users who use a system are revealed with the UML. The restrictions, if any, of a system are also exposed with the UML [57, 58].

Model checking is called an automatic technique for verification and validation of the specification of the model. It is used to verify the finite state in a reactive or distributed system. Specifications are stated to be expressed in temporal logic to model as a state transition graph in a distributed system. Both Clark [59] and Alen Emerson [60] initially developed this technique in 1981. It is used for validation, verification, and is also appropriate in logical formalism, such as

temporal logic, and it can also be used to check the validity of the property in a reactive or distributed system. It is a general approach that is used in different areas, such as software engineering and hardware verification. There are three types: first is model checking linear, second is branching, and third is real-time (branching) of temporal logic. The first two types of model checking are used for qualitative or functional aspects of the system, whereas the third type of model checking is used additionally in some quantitative analysis of the system. There are also two methods used for model checking: the first is logic-based or heterogeneous, and the second is behavior-based or homogeneous. Let us assume the behavior of the reactive system for determining the Boolean ‘x’ state variables ($s_1, s_2, \dots, s_n$). So, a transition relation of the given system can also be expressed as a (Boolean formula). $T(s_1, s_2, \dots, s_n, s'_1, s'_2, \dots, s'_n)$ where “ $s_1, s_2, \dots, s_n$ ” represents the current state in the reactive system, whereas “ $s'_1, s'_2, \dots, s'_n$ ” represents the next state in the reactive system of a Boolean formula [59–63].

In this study, the researchers have explained that our lives are gradually free with smart devices tied to the internet, frequently called the “IoT”. But as these devices rise in number and gather sensitive data, securing them becomes essential. The visible environment of the internet is prone to attacks, and vulnerabilities can be horrible concerns for both users and the devices themselves. The rapid progress of the IoT market often ranks quickness over security, leaving the industry requiring a broader outlook. After all, smartness does not promise safety, and identifying potential flaws early in the design phase is significant. Formal verification is a way to mathematically prove or disprove if a system behaves as intended. Two common approaches are theorem proving and model checking. Both can be applied early in design. Theorem proving uses logic and math to create formulas representing system behavior. Model checking, on the contrary, verifies if a property holds true for a model of the system’s implementation. This article discovers the present state of formal methods, including tools, languages, and communication protocols, all aimed at securing the IoT [64, 65].

In the research work of Hofer-Schmitz and B. Stojanović [66], they presented a detailed, comprehensive review of formal verification approaches for various IoT properties. The research highlights the status of formal verification in fast detecting weaknesses and offers involved kindnesses of various properties and practices. The work outlined in this study explains four main key areas of application,

that is, functional assessments, security property assessments, suggestions to improve frameworks with early established security property duties, and protocol implementation validations. Furthermore, the document deliberates security attributes and protocol utilities, combined with consideration of common model assessments and the specific challenges to IoT, comprising their constraints.

A recent survey [67] presents a complete review of formal verification methods used in various IoT applications, including manufacturing, smart grids, industry, robotics, and enterprises. It associates three main methods: first is model checking, second is theorem proving, and the third is process algebra. The study also analyzes the strengths and weaknesses of different formal verification practices in IoT environments. This study presents the development of an intelligent desk lamp that automatically adjusts the brightness to free sightless areas. In this research, they use a convolutional neural network algorithm to recognize objects in the workspace and track their location, so that they are always as good as possible without leaving a particular spot in the darkness wherever you put it for light. The smart desk lamp was compared with other models driven by the support vector machine and Back Propagation Neural Network (BPNN) during the evaluation phase. The experiments indicate that the proposed smart pole-lamp based on Convolutional Neural Network (CNN) edges out other comparable devices with considerable target recognition and positioning improvement. As tracking became constant, the lamp accurately moved to a position where it had minimal positional faults and was, therefore, capable of quickly following the movement of targets to provide the universal image lights. The novelty of this work is its new application of the CNN algorithm on visual tracking in a smart desk lamp that enables the lamp to change its lighting angle and position automatically as well. It allows users to make the best use of the environmental light environment to minimize the requirement for manual operations and to ensure that lighting is sufficient for specific activities while offering increased visual comfort and productivity. The technology might also be utilized more commonly in other situations that require exact, adaptable lights, for taking pictures in industrial workspaces or in the medical field. The spotless combination of CNN in the smart desk lamp sets a groundbreaking model for improved lighting solutions, indicating the emergence of even more intelligent, flexible, and consumer-friendly lighting methods on the road [68].

This study supports the use of intelligent systems in building-wise design architecture, where its emphasis is on energy utilization. This area involves advanced energy-saving approaches and environmental design principles that need well-thought-out management decisions to run a positive project. For observation of energy efficiency, key performance values are engaged to establish the project risk and cost adjustment and schedule variation that facilitates verifying project improvement on building smartness. Furthermore, this study compared building information modeling with traditional decision-making processes and displayed examples of these processes, not providing data on performance deviations. The Delphi study identified 19 significant construction cost variables, with the strength of those revised factors in decreasing order was building number of floors, structural design type, and floor round size. Analysis of multicriteria decision-making exposed that the one on which their current system was based permits better prediction of project costs and schedules; large differences between planned and actual results were also identified in this research work [69].

Mostly, the drivers face a problem when they are looking for available locations. Inadequate parking accessibility cannot provide residents with enough spots to park their cars. Drivers now waste time continuously looking for vacant areas; this has significant implications in the way that we move about by causing drivers to be delayed to appointments, late on their bills, or even detained at hospitals. Though several researchers have established parking management systems, few guide vehicles in real time. The above study proposed a smart parking guidance system based on IoT-enabled Raspberry Pi and Android applications with a k-nearest neighbors' algorithm for vehicle location. Using fingerprinting in Wi-Fi signal strength can navigate and detect parking in real-time. This system makes use of Raspberry Pi hardware, is connected to sensors, and uses the ThingSpeak IoT cloud that allows for parking monitoring in real time. The Android application can navigate to show the available parking with real-time location evaluation and turn-by-turn route supervision. The sensor data is managed via Python, and then it uses the message telemetry transport protocol to send the data to the cloud, where this data is displayed across an Android application so the user can navigate toward free slots with high accuracy. Using the weighted k-nearest neighbors' algorithm with this unique system, the accuracy of the positioning was calculated for an average error, and this also enhanced the accuracy of the proposed system [70].

a. Why temporal logic of action language (TLA+) and formal methods

TLA+ is basically a framework developed for helping software engineers write great, rigorous systems, particularly which supervise to be complex and have many parts that need to play with in concert with one another. It is like creating an intricate blueprint for a house and wanting all the pieces to fit together before beginning construction. There are a pair of very important reasons why we use TLA+ and formal methods in general:

Early detection: Using formal methods makes it less likely that we will design with mistakes since they can catch problems before any real code is written. That is so much cheaper and easier than finding issues after it's been built. Amazon discovered critical bugs in its AWS services using TLA+ that would have been very hard to find through traditional testing [11, 71].

Mathematical as surety: Rather than just testing a system and praying all the issues are fixed, TLA+ allows us to prove mathematically that some bad things cannot happen. This is especially important for critical software like medical devices or financial software, where failures can be dreadful.

Complexity: Modern distributed systems (e.g., cloud services) can have many moving parts, all working together. TLA+ can help us understand and verify these interactions.

Find edge cases testing might miss: A TLA+ specification excels at finding edge cases that traditional testing could easily miss.

Better communication: Writing a TLA+ specification is like producing an exact description of how the system should behave. This way, team members in the context will understand the system better, and there will be no bugs from misunderstanding. Using TLA+ when designing a smart courtroom system is a remarkable connection between logical procedure, computer science, and formal verification.

b. Challenges in smart building

The smart courtroom systems are yet more challenging because they have a terrifying state space complication. Sense of all the people who participate in a normal court event: a judge, lawyers, witnesses, jurors, clerks, and even the public. Every member has its role, permissions, and so on, which makes a massive number of possible system states. This difficulty reveals itself in the need to carefully govern state transitions while ensuring system reliability and fulfilling all constraints when designing with TLA+.

The formal specification also includes extra density related to security and privacy considerations. As well, transparency, where law permits, shall continue to hold up while access controls strengthen. This dual nature forms a fundamentally interesting challenge in formal verification, where it is necessary to prove that no unauthorized access can ever be gained but that legitimate access can always be obtained. These requirements need to be captured in the formal specifications in a way that can be verified mathematically and still make sense for use in a practical implementation. A further challenge is the need for real-time performance. In court procedures, the need for a quick response and live information feed is commonly necessary. The context of TLA+, modeling these temporal aspects, parsing between safety and liveness properties is delicate to define when a system should do its work, making it not only able to correctly execute but also timely with respect to all aspects. The formal specification must cover network delays, possible failures, and requirements such as all parties having the same view of the system state. Both the specification complexity and the verification process are considerably affected by this distributed aspect.

Properties: There are a significant number of requirements that must be collected to enable the smart courtroom system for property verification. Safety properties are things that make sure nothing bad happens; nothing gets anywhere without rights, no records get lost, and no procedure-breaking happens. A liveness property ensures that eventually, good things happen (i.e., cases move forward as needed, participants get their chance to do what they must do, and the process moves to a conclusion). These properties are required to be checked in a formal fashion and often involve complex proof techniques and careful handling of the state space explosion problem.

Design challenges: Adopting a design also presents its own challenges. We need to carefully walk the path from formal specification to practical implementation if we want a critical property verified on one level of abstraction, retained in one that is closer to hardware, where it will eventually be running. Dealing with physical world problems, such as hardware failure, network partitions, and humans making mistakes, yet still maintaining the guarantees we were given from formal verification. Using AI and machine learning in smart courtroom systems makes the verification more difficult. These components, which are spelled out informally and done for activities, such as text generation or document understanding, need to be embedded as part of a

formal specification in a manner that enables the preservation system to be unchallengeable while accounting for their probabilistic nature. It is an innovative part of formal method research, and the number of AI components that should be processed efficiently will require adaptation to traditional verification techniques. The challenges in this area will be determined by further changes in requirements and technological developments. Formal specifications should be given some relaxation, with the flexibility of being rewritten without letting degradation into a lack of assertion, and properties, once verified, should remain verified. Achieving properly verified smart courtroom systems in procedure thus requires a balanced view that combines a grounded set of practical considerations with theoretical methods. It requires formal methods and software engineering knowledge, as well as knowledge of legal procedures and court operations. So, there is a lot to be positive about, and this remains an important area for future research & development.

III. Proposed Framework

This section of the study represents our proposed approach. In designing software research, various methods are used to specify, model, verify, and validate the system. This includes applying TLA+ specification language. See Figure 1 that presents the proposed system process in the SmartCourt TLA+

module. In Figure 2, the inputs and outputs of an SMCBS are represented, while Figure 3 presents the entities within the system, corresponding to the system variables in the SmartCourt TLA+ module.

a. Unified Modeling Language

The UML stands out as the foremost language to show the abstract model of any system. Offering graphical notations, it adeptly captures the properties of a system [57, 58]. The system received the inputs and feedback from three different types of sensors: smoke, light, and temperature sensors. Additionally, users, including presiding officers, court staff, advocates, and litigants, directly provide input through their logins, applying their unique usernames and passwords. The system's outputs convey actions such as correct or incorrect logins, temperature status within the building, smoke detection alerts, and light brightness levels. Figure 2 demonstrates an activity diagram for the proposed smart courtroom building, showcasing all sequential actions executed by the system and enhancing overall comprehension. Initially, accused, employees, and visitors enter the court building, every retaining a separate username and password. Upon entering the correct username and password, the accused and employees gain access through the main door, which automatically opens for them. Visitors missing login credentials must pursue approval from the reception employee; if

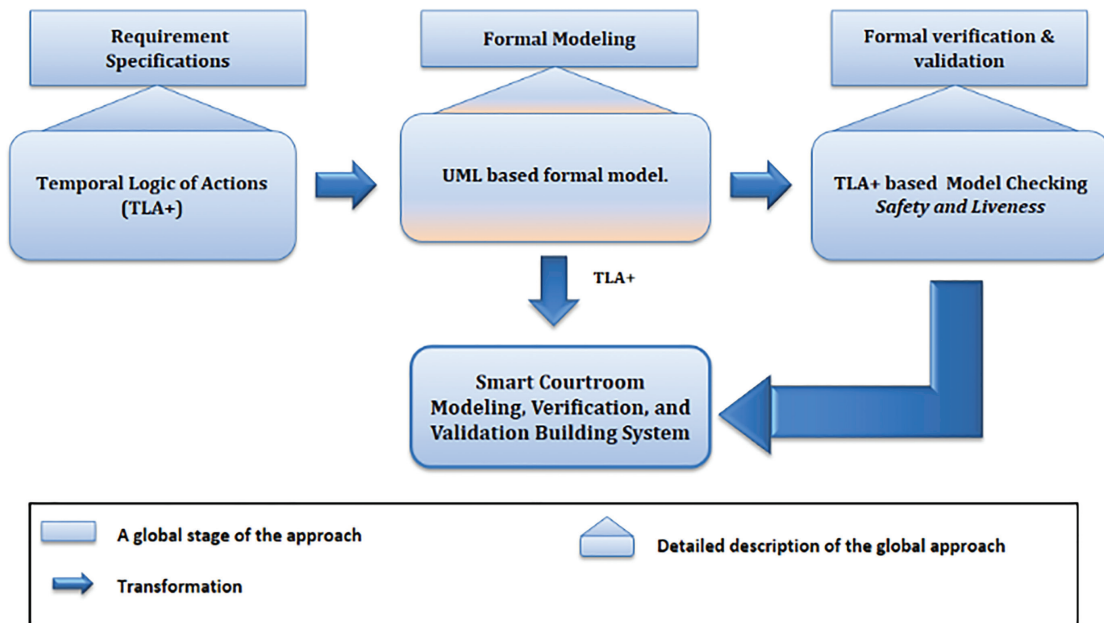


Figure 1: Proposed approach.

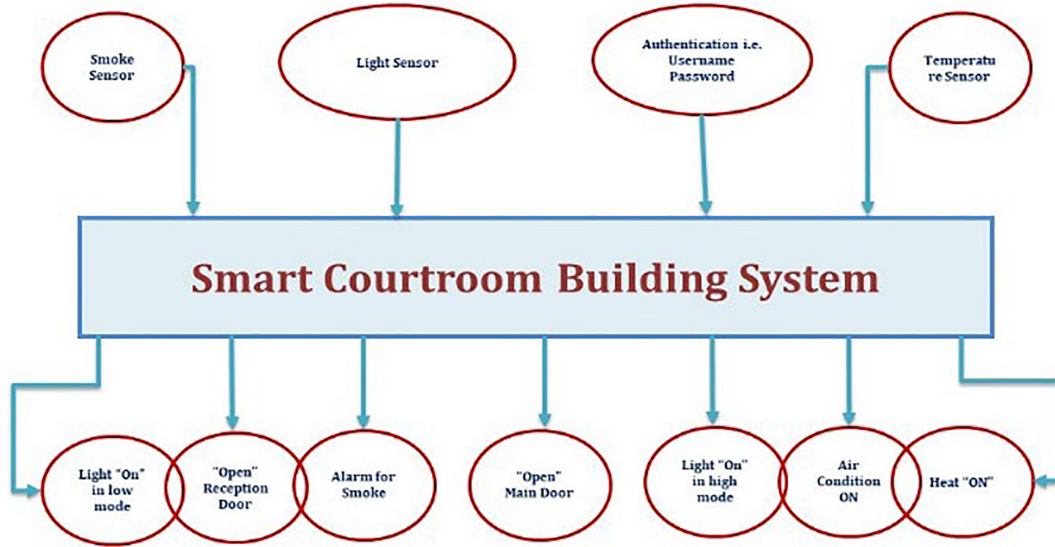


Figure 2: Inputs/outputs of SMCBS. SMCBS, smart courtroom building system.

approved, only the reception door grants them entry, while all doors remain closed if not accepted. After the building is vacant, numerous systems are automatically deactivated. However, upon the arrival of a student or employee, all subsystems initiate automatically. A temperature sensor monitors the building's temperature, triggering the heating system if it falls below a specified threshold (20°C) and activating the AC system if it exceeds another threshold (23°C), as we assumed the temperature as per the given atmosphere of the country. Additionally, the lighting subsystem adjusts according to ambient sunlight to minimize power consumption. For instance, on sunny days, the system switches to low-light mode since bright lights are unnecessary. Meanwhile, the burn discovery system operates continuously to ensure the safety of the building. If smoke or fire is detected, alarms sound, and in the event of a fire, all doors automatically open, facilitating a swift and secure evacuation for everyone within the building.

b. TLA+ formal specification

This portion of the study includes the formal specification of our proposed smart courtroom building model, after deployment of the TLAplus (TLA+) specification language. We have also considered the points during the writing of the specification, which are represented in Table 2. As the automation of the system depends on a diverse collection of sensors, and the use of TLA facilitates the representation of

all system operations, in Figure 3, the inputs and outputs of an SMCBS are represented, while Figure 2 presents the entities within the system, corresponding to the system variables in the SmartCourt TLA+ module.

The top figure of the module of Smart Courtroom structures the variables defined in Figure 4. After the name of the module, following keyword `EXTENDS` is used to refer to libraries, such as `Naturals`, `TLC`, `Integers`, and `Sequences`. User variable serves to identify individuals authorized to enter into the courtroom building, including presiding officer/judge, court employees/staff, advocates, and litigants.

The visitor and user variables represent an individual missing a username or password, requiring approval from an employee to access the building. The `door_variable` indicates the core entry door for the presiding officers and court staff, while the `reception_door` describes the entrance for advocates and litigants. The `alarm_variable` links to the smoke detector, activating an alarm upon smoke detection. Username and password variables store the unique login credentials for each presiding officer and court employee. The thermostat variable captures the sensed temperature, and the AC and Heat variables represent the heating and air conditioning systems. The lighting system comprises four variables first `light` indicates the on or off status, the second `light_status` receives input from a sensor determining sunlight conditions, producing output for either `light_high mode` (bright lights on cloudy

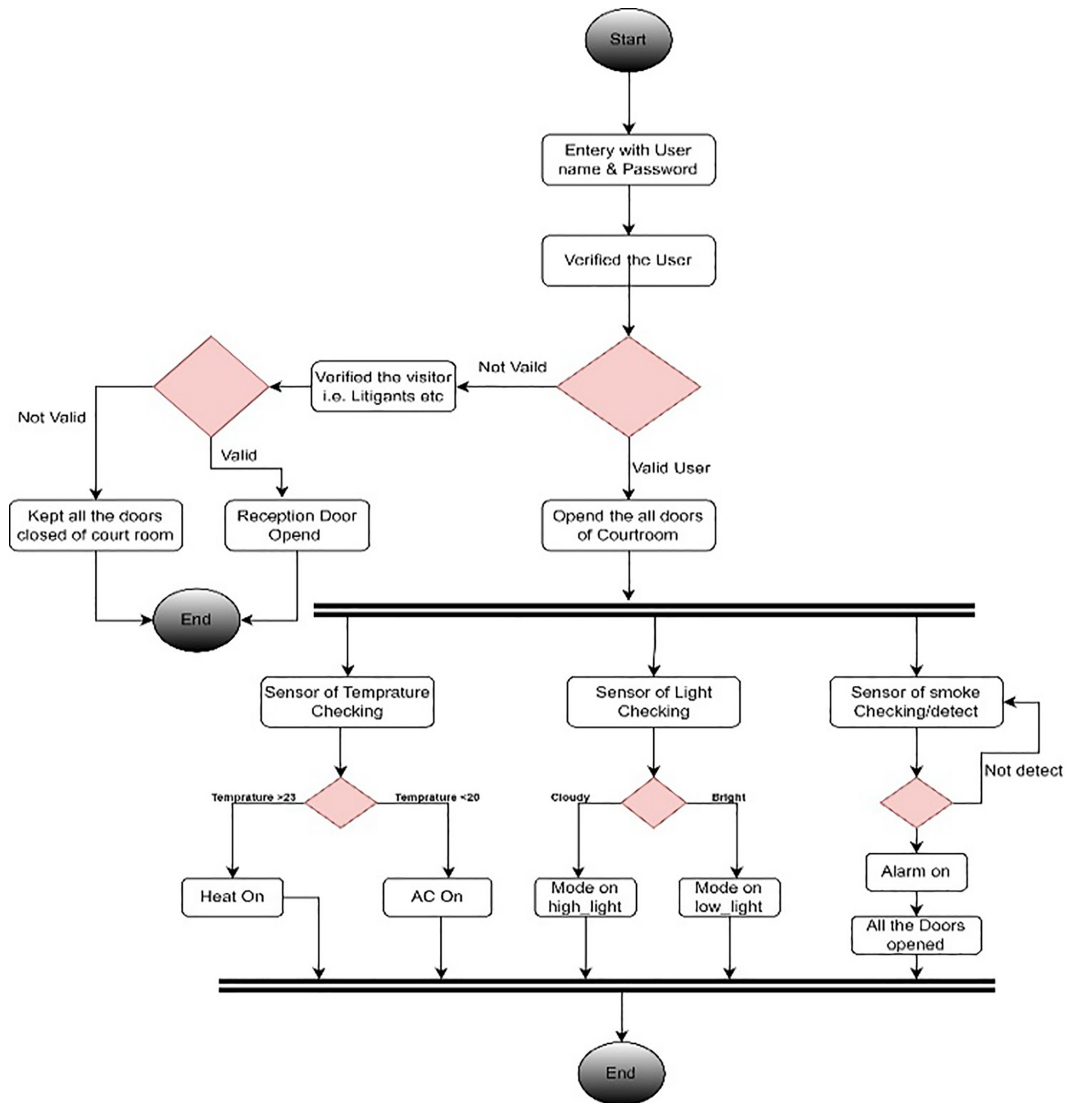


Figure 3: Smart courtroom activity diagram.

days) or light_low mode (dim lights on sunny days for energy efficiency). So, system invariants are well-defined below, including conditions of user having four possible values (1: presiding officer, 2: court employee, 3: advocates, and 4: litigants), presiding officers requiring username and password verification to enter the judicial complex building, litigants and advocates needing permission for entry, and temperature remaining stable within the range of 20°C–23°C as we assumed earlier in our case. We categorized users into four types (presiding officer, court employee, advocates, litigants) based on typical courtroom hierarchies. This simplifies access control but may not capture the full complexity of real-world court operations.

However, the specified conditions to explore the function are detailed below. So, within any TLA+ module, it is necessary to state the invariants, the Init function, along with the Next function. Invariants describe the conditions and restrictions in a given system, while the Init function explains the preliminary values of the system's variables, and the Next function allocates the next-state action. Figures 5–10 show different functions, specifications, and requirements in TLA+. The EnterCourtroom function facilitates the entry of litigants and employees into the courtroom, requiring user type, username, and password as inputs. Valid users gain entrance through the doors upon execution.

The VerifyVisitor function authenticates and verifies visitors to the courtroom, designating six types

of visitors and permitting entry solely through the reception door for authorized visitors. The SetLight function, activated upon the arrival of the first student or employee, adjusts light brightness based on input from the light sensor to optimize power consumption. The light conditions are set for their impact on energy efficiency, which is based on general principles of smart lighting

Function SetTemperature regulates the building's temperature within the range of 20°C–23°C across all seasons, utilizing input from the temperature sensor while temperature has earlier assumed. This also maintains the heat and cold in the courtroom according to the building temperature. For enhanced safety, a smoke sensor informs the Detect_Smoke function. In case of fire or smoke, the alarm activates, and all doors open, enabling an immediate exit from the courtroom in the building. Temperature range (20°C–23°C): this range was chosen based on standard comfort levels for courtroom environments. However, this assumption may not account for regional variations or personal preferences. A wider or narrower range could impact energy consumption and user comfort.

However, the Next function in TLA+ plays a key role by collecting and executing whole functions to switch to the next state in a given system. The Spec_ function in TLAplus directs the execution of all system specifications, signifying the main run of the system. This represents the ending statement in a system's specification, subsequently subject to verification through TLC, the TLA+ model checker.

c. TLC model checker in the toolbox for verification of the specification

Once the specification has been successfully parsed, the smart courtroom building module can

Table 2: Information of specification in TLA+

Type name	Value
Total number of variables	21
Extends library variables	04
Total number of initializations	21
Total number of off events	06
Total number of on events	06

be executed after using the TLC model checker, as the same is built-in TLA+ tool. For a successful run of the TLC model checker, two parameters must be configured for the model overview after its creation. So, the first contains selecting the initial-predicate in the toolbox of TLA+ and determining the behavior of the next-state relation, while the second entails unchecking the deadlock option. Once all parameters are set in accordance with the specifications shown in Figure 13, the model checker can be executed. If the specification is free of errors, then the TLC model checker will adeptly parse all states as defined in the specification using the TLA+ toolbox. Otherwise, it will fail to show any distinguished states in both cases of the Init and next state. So, this process is completed by the distinguished states to couple with successful exploration of the state space, serving as evidence of the system's correctness.

d. Explanation of system behavior and specification

TLA+ is used for obtaining the system behavior and specifications of an SMCBS as follows: -

Variables and state representation: In our proposed research, TLA+ is used to define system

```

MODULE SmartCourtroom
EXTENDS Naturals, TLC, Integers, Sequences

VARIABLES user, visitor, presiding-officer, court-staff, lawyers, litigants,
           door, reception-door, alarm, temprature, light, smoke_detector,
           username, password, AC, Heat, light_high, light_low, light_status,
           courtroom, Enter

```

Figure 4: Variable declaration in the smart courtroom module.

variables (21 in total) that represent different aspects of the system state, such as user, visitor, presiding_officer, court_staff, lawyers, litigants, door, reception_door, alarm, temprature, light, smoke_detector, username, password, AC, Heat, light_high, light_low, light_status, and enter, as shown in Figure 4. Another step for the system behavior and specifications is the Initial state specification, as it is the Init function in which TLA+ is used to define the initial values of all system variables shown in Figure 6. Another step that is involved in system behavior modeling for explaining the several functions in TLA+ that model different

behaviors of the system, such as EnterCourtroom, VerifyVisitor, SetLight, SetTemperature, and DetectSmoke, as shown in Figures 7–9. The next step is also compulsory for any TLA+ specification for running specification and model verification, and it is the Next-state relation function that combines all the individual behavior functions to define how the system transitions from one state to another. The system specification Spec function in TLA+ indicates the whole system specification, after combining the initial state and next-state relation, as shown in Figure 11. We have used the TLC Model Checker for

$$\begin{aligned}
 \text{TypeOK} \triangleq & \wedge \text{user} \in 1 \dots 6 \\
 & \wedge \text{username} \in \{\text{"correct"}, \text{"wrong"}\} \\
 & \wedge \text{password} \in \{\text{"correct"}, \text{"wrong"}\} \\
 & \wedge \text{door} \in \{\text{"open"}, \text{"closed"}\} \\
 & \wedge \text{reception_door} \in \{\text{"open"}, \text{"closed"}\} \\
 & \wedge \text{alarm} \in \{\text{"ON"}, \text{"OFF"}\} \\
 & \wedge \text{litigants} \in \{\text{"valid"}, \text{"invalid"}\} \\
 & \wedge \text{light} \in \{\text{"ON"}, \text{"OFF"}\} \\
 & \wedge \text{light_high} \in \{\text{"ON"}, \text{"OFF"}\} \\
 & \wedge \text{light_low} \in \{\text{"ON"}, \text{"OFF"}\} \\
 & \wedge \text{light_status} \in \{\text{"Bright"}, \text{"Cloudy"}\} \\
 & \wedge \text{smoke_detector} \in \{\text{"discover"}, \text{"not_discover"}\} \\
 & \wedge \text{temprature} \in 20 \dots 23 \\
 & \wedge \text{Heat} \in \{\text{"ON"}, \text{"OFF"}\} \\
 & \wedge \text{AC} \in \{\text{"ON"}, \text{"OFF"}\}
 \end{aligned}$$

Figure 5: Smart courtroom TypeOk.

$$\begin{aligned}
 \text{Init} \triangleq & \wedge \text{door} = \{\text{"closed"}\} \\
 & \wedge \text{reception_door} = \{\text{"closed"}\} \\
 & \wedge \text{alarm} = \{\text{"OFF"}\} \\
 & \wedge \text{temprature} = 20 \\
 & \wedge \text{light} = \{\text{"OFF"}\} \\
 & \wedge \text{AC} = \{\text{"OFF"}\} \\
 & \wedge \text{Heat} = \{\text{"OFF"}\} \\
 & \wedge \text{light_high} = \{\text{"OFF"}\} \\
 & \wedge \text{light_low} = \{\text{"OFF"}\} \\
 & \wedge \text{password} = 0 \\
 & \wedge \text{visitor} = 6 \\
 & \wedge \text{user} = 1 \\
 & \wedge \text{smoke_detector} = \{\text{"not_discover"}\} \\
 & \wedge \text{username} = \{\text{"sss"}\} \\
 & \wedge \text{light_status} = \{\text{"Bright"}\}
 \end{aligned}$$

Figure 6: Init function of the smart courtroom building system. SMCBS, smart courtroom building system.

$$\begin{aligned}
\text{Very Visitor} &\triangleq \bigvee \wedge \text{user} = 5 \\
&\quad \bigvee \wedge \text{visitor} = \{\text{"valid"}\} \\
&\quad \wedge \text{reception_door} = \{\text{"open"}\} \\
&\quad \wedge \text{door} = \{\text{"closed"}\} \\
&\quad \wedge \text{user} = 6 \\
&\quad \wedge \text{visitor} = \{\text{"not_valid"}\} \\
&\quad \wedge \text{reception_door} = \{\text{"closed"}\} \\
&\quad \wedge \text{door} = \{\text{"closed"}\} \\
\\
\text{SetLight} &\triangleq \bigvee \quad \wedge \text{door} = \{\text{"open"}\} \\
&\quad \wedge \text{light_status} = \{\text{"Bright"}\} \\
&\quad \wedge \text{light_low} = \{\text{"ON"}\} \\
&\quad \wedge \text{door} = \{\text{"open"}\} \\
&\quad \wedge \text{light_status} = \{\text{"Cloudy"}\} \\
&\quad \wedge \text{light_high} = \{\text{"ON"}\}
\end{aligned}$$

Figure 7: Verification of visitors/litigants and SetLight functions.

$$\begin{aligned}
\text{SetLight} &\triangleq \bigvee \quad \wedge \text{door} = \{\text{"open"}\} \\
&\quad \wedge \text{light_status} = \{\text{"Bright"}\} \\
&\quad \wedge \text{light_low} = \{\text{"ON"}\} \\
&\quad \wedge \text{door} = \{\text{"open"}\} \\
&\quad \wedge \text{light_status} = \{\text{"Cloudy"}\} \\
&\quad \wedge \text{light_high} = \{\text{"ON"}\}
\end{aligned}$$

Figure 8: SetLight function of SMCBS.

verification that is built into the TLA toolbox built into the TLC model checker. First, we have parsed the model specification to ensure its syntax is correct, as shown in Figure 12. Later, we set up the model in which parameters are configured in the model overview, including choosing the initial predicate and next-state relation, as shown in Figure 13. The next step, which is involved in the verification, is state space exploration, where the TLC model checker explores all the possible states of the system model as defined by the specification. The next step involved correctness verification, and the model checker verifies that the system behaves correctly according to its specifications by exploring all possible states and checking

for invariant violations during the model checking process method. Finally, we created the report regarding the statistical analysis. The TLC model checker provides a report describing the number of distinct states explored, which serves as evidence of the system's correctness.

e. Security measures

The access control for the accused, lawyers, litigants, court staff, and presiding officers represents the system using usernames and passwords for authentication. The accused and employees (including presiding officers, court staff, and lawyers) have

$$\begin{aligned}
SetTemperature &\triangleq \vee \quad \wedge door = \{ "open" \} \\
&\quad \wedge \text{ IF } temprature \Rightarrow 20 \text{ THEN } Heat = \{ "ON" \} \\
&\quad \text{ ELSE } Heat = \{ "OFF" \} \\
&\quad \quad \wedge door = \{ "open" \} \\
&\quad \wedge \text{ IF } temprature = 23 \text{ THEN } AC = \{ "ON" \} \\
&\quad \text{ ELSE } AC = \{ "OFF" \} \\
DetectSmoke &\triangleq smoke_detector = \{ "discover" \} \\
&\quad \wedge alarm = \{ "ON" \} \\
&\quad \quad \wedge door = \{ "open" \} \\
&\quad \quad \wedge reception_door = \{ "open" \} \\
&\quad \quad \wedge alarm = \{ "OFF" \} \\
&\quad \quad \wedge smoke_detector = \{ "not_discover" \}
\end{aligned}$$

Figure 9: SetTemperature and DetectSmoke functions in SMCBS.

$$\begin{aligned}
DetectSmoke &\triangleq smoke_detector = \{ "discover" \} \\
&\quad \wedge alarm = \{ "ON" \} \\
&\quad \quad \wedge door = \{ "open" \} \\
&\quad \quad \wedge reception_door = \{ "open" \} \\
&\quad \quad \wedge alarm = \{ "OFF" \} \\
&\quad \quad \wedge smoke_detector = \{ "not_discover" \}
\end{aligned}$$

Figure 10: DetectSmoke functions in SMCBS.

obtained their unique login credentials. Afterwards, to enter the correct username and password, the main door of the building automatically opens for them. On the contrary, the approval process for visitors is also compulsory for those with no login credentials and must seek approval from the reception court staff or employees. If approved, only the reception door permits them entry for entrance into the building. If not approved, then all doors remain closed to the visitor. The doors of the building system have separate variables for the main door at the entrance point of the building (for court staff, presiding officers, and employees) and the reception door (for visitors, litigants, and lawyers). The doors are handled to be controlled automatically based on verification

and approval status by the reception staff. Safety and security during smoke detection continued to operate with the subsystems for the building. If any smoke is detected, then alarms are set in the building area, and when found or detected fire then automatically all doors open for safety to facilitate the swift evacuation of the people during the alarm situation in the building.

IV. Results

In this study, we have built and developed a TLA+ representation of SMCBS by examining security breaches at both the system and fundamental levels. The TLC model checker's statistical

analysis specifies the end of state space exploration. However, the report details the count of separate states in both initial (Init) and consequent actions. Specifically, the number of distinct states for “Init” is 1, for Next is, and the total distinct states are equivalent to the sum of “Init” and “Next”. Resultantly, the overall states generated by the TLC model checker for the SMCBS specification. The statistical findings and report confirm the TLC model checker’s capability to manage a significant number of states. The formal specification’s accuracy is validated by the total sum of distinguished states with the help of the TLC model

$$\begin{aligned}
 \text{Next} &\triangleq \begin{aligned} &\vee \text{Enter} \\ &\vee \text{SetLight} \\ &\vee \text{SetTemperature} \\ &\vee \text{DetectSmoke} \end{aligned} \\
 \text{Spec} &\triangleq \text{TypeOK} \wedge \text{Next}
 \end{aligned}$$

Figure 11: Next and spec functions in SMCBS.

checker, as demonstrated in a generated statistical report of the TLA+ toolbox.

a. Several additional security features with verification and validation

There are several additional security features, such as role-based access control for implementation at different access levels for various user types. The judges and court staff can enter all areas of the court building with unique login credentials. On the contrary, other user types, such as accused, litigants, lawyers, and any type of visitors, can also enter specific areas of the building after using login credentials or obtaining permission from the security permission office. The time mechanism for entering the building is fixed according to the standard for the building and other types of users, and they can only be entered into specific areas of the building.

Verification and validation process using the TLC model checker: We also have the security properties that are formally specified for our proposed system as TLA+ predicates.

No unauthorized access: $\forall u \in \text{Users}: u.\text{location} \neq \text{“restricted_area”} \vee u.\text{clearance} \geq \text{“high”}$

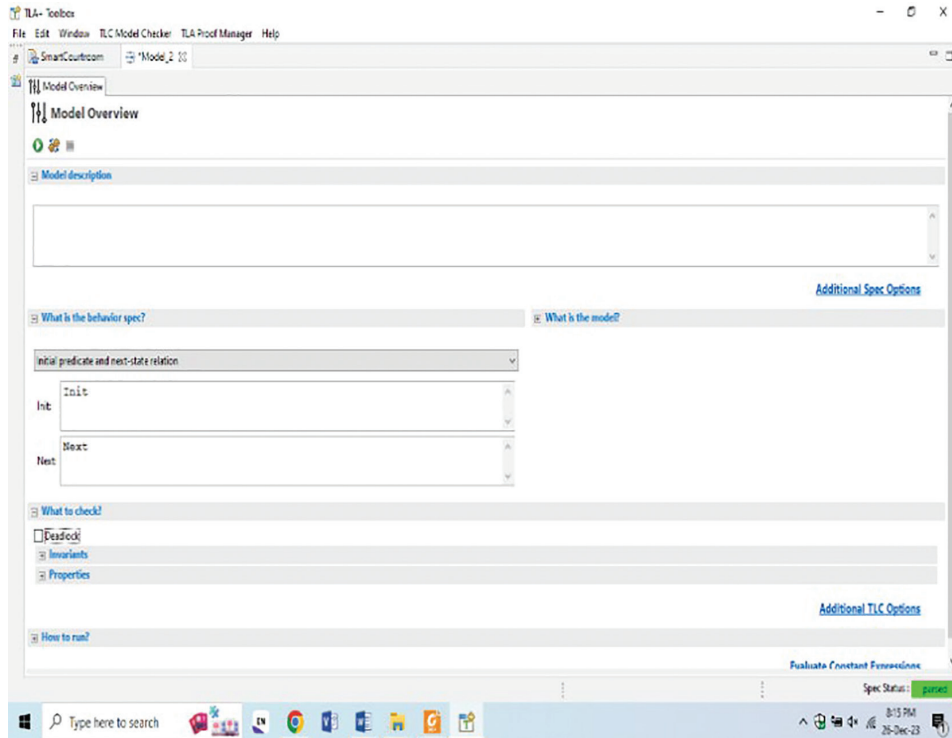


Figure 12: Parsed model of TLA+ specification.

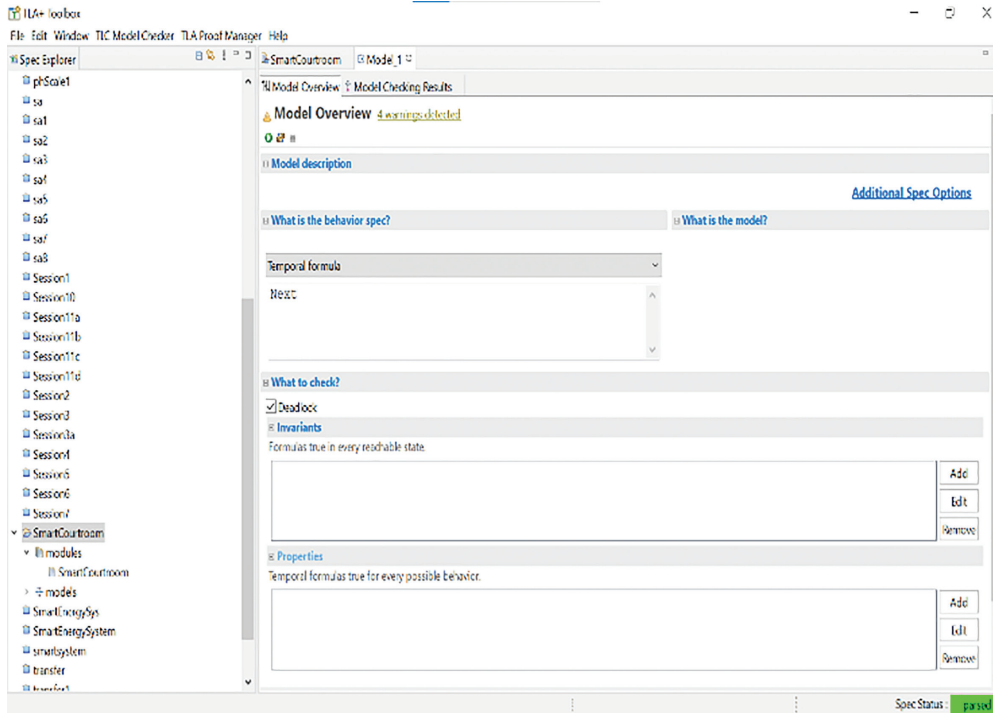


Figure 13: Parameter setting in model overview.

Proper authentication: $\forall u \in \text{Users}: u.\text{authenticated} \vee u.\text{provided_credentials} = u.\text{stored_credentials}$

After running the TLC model checker, we explored all possible states of the system. We have verified the invariants to ensure that the system is secured and must always hold true for the system.

- NoUnauthorizedAccess
- ProperAuthentication

we have also used the temporal properties that describe how the system should behave over time.

- AttemptedBreak $\Rightarrow \Diamond \text{Alarm}$ (Always, an attempted break-in leads to an alarm eventually).

V. TLA-Plus Model Analysis

This section is related to the process of verification using the TLC model Checker. Some details are given for understanding the process of the TLC model checker. The process of using the TLC model checker to verify and validate the system model involves the following: We have created a formal model of the building system using TLA+ and PlusCal languages. We have

also defined system properties and specifications in TLA+. We have configured the TLC model checker in the TLA+ toolbox, including setting initial predicates and next-state relations. Afterward, we run and compile the TLC model checker. We have also analyzed the results, including the count of distinct states explored. We have mentioned that if the specification is free of errors, the TLC model checker will parse all states as defined in the specification. If there are any errors, it will fail to show any distinguished states.

UML during verification and validation: We have used the UML for the purpose of modeling our conditions, and we have written our specifications as per the requirements of the court building. Afterwards, we verified the invariant and temporal properties for the validation of our results. The UML diagram used here is only for the modeling of our conditions to some extent. We have specified the property specification to define the correctness of properties in TLA+ based on the expected behavior represented in UML diagrams. We have also checked the model specifications to run the TLC model checker on the TLA+ specification. We have also checked the Error detection. If TLC detects violations of specified properties or inconsistencies in the model. As previously mentioned, several tools are available for system

specification, including VDM-SL, Z, RAISE [72], and TLA+ [49–51, 73]. For the verification of our system, we chose to utilize TLA+. TLA+ stands out for its enhanced capabilities, particularly its model-checking technique, which ensures the verification of a system. Using the TLA+ toolbox, we enhanced the correctness of our proposed system. The parsed model, as depicted in Figure 12, confirms the accurate interpretation of the proposed model using the TLA+ toolbox and shows a green box at the bottom left corner of the given, and has undergone correct parsing through the TLA+ toolbox, setting the stage for verification through the TLC model checker. The mathematical analysis of this proposed model shows the expansion of TLA+ techniques, and the correctness of specifications and checks has been validated. Additionally, TLC has verified the proper use of object variables in the model.

VI. Conclusion and Future Work

This study introduces a smart and secure courtroom building system, modeled through a combination of UML and TLA+. UML explains system apparatuses and the sequence of actions, while TLA+ captures the system's behavior and specifications. Verification of system properties has been achieved by extending TLA+ and its toolbox, employing model parsing to understand the system's performance. The proposed system is considered to collect data from various sensors, temperature sensors for building temperature, light sensors to optimize power consumption, and smoke detectors for fire safety. Additionally, the system enhances security by allowing access only to the accused and employees with valid usernames and passwords, requiring visitors to obtain approval from a courtroom employee for entry.

In future enhancements to the system, we will focus on two aspects: first, additional features, such as notably safety, access level security, and others, will be incorporated in recognizing the paramount importance of models of the system. Second, we will design to verify and validate our present research with more elaborate models, establishing its correctness through the TLC model checker. This process will also include indicating examples of incorrect or contrary UML features, illustrating how the TLC model checker can identify these errors for correction before implementation. We will also include more formal languages to enhance further work. Furthermore, there is also a need to work more on this smart system with more specification languages, such as VCC, B, Z, and VDM, in near future research work.

References

- [1] R. S. Kumar, P. L. Pugalthanthi, N. Dhanyaa, V. Kaviyanjali, K. Dinesh, and K. Kathirvel, "IoT-based smart monitoring topologies for energy-efficient smart buildings," in *Modern Computational Techniques for Engineering Applications*: CRC Press, 2024, pp. 79-106.
- [2] E. Kamel, S. Habibi, and A. M. Memari, "State of the practice review of moisture management in residential buildings through sensors," in *Structures*, 2024, vol. 59: Elsevier, p. 105698.
- [3] M. Tunzi, T. Benakopoulos, Q. Yang, and S. Svendsen, "Demand side digitalisation: A methodology using heat cost allocators and energy meters to secure low-temperature operations in existing buildings connected to district heating networks," *Energy*, vol. 264, p. 126272, 2023.
- [4] G. Liu, H. Meng, G. Qu, L. Wang, L. Ren, and H. Lu, "Real-time monitoring and prediction method of commercial building fire temperature field based on distributed optical fiber sensor temperature measurement system," *Journal of Building Engineering*, vol. 70, p. 106403, 2023.
- [5] D. Wang, Y. Chen, W. Wang, C. Gao, and Z. Wang, "Field test of Model Predictive Control in residential buildings for utility cost savings," *Energy and Buildings*, vol. 288, p. 113026, 2023.
- [6] A. Dennis, B. Wixom, and D. Tegarden, *Systems analysis and design: An object-oriented approach with UML*. John Wiley & sons, 2015.
- [7] A. Clark and A. Evans, "Foundations of the Unified Modeling Language," in *Proceedings of the 2nd Northern Formal Methods Workshop.*, 1997: Springer.
- [8] É. André, S. Liu, Y. Liu, C. Choppy, J. Sun, and J. S. Dong, "Formalizing UML State Machines for Automated Verification—A Survey," *ACM Computing Surveys*, 2023.
- [9] C. Zhang, B. Li, T. N. Sainath, T. Strohman, and S.-y. Chang, "UML: A Universal Monolingual Output Layer For Multilingual Asr," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023: IEEE, pp. 1-5.
- [10] S. Tasiran, Y. Yu, B. Batson, and S. Kreider, "Using formal specifications to monitor and guide simulation: Verifying the cache coherence engine of the Alpha 21364 microprocessor," in *Proceedings of the 3rd IEEE Workshop on Microprocessor Test and Verification, Common Challenges and Solutions*, 2002.
- [11] C. Newcombe, "Why amazon chose TLA+," in *International Conference on Abstract State Machines, Alloy, B, TLA, VDM, and Z*, 2014: Springer, pp. 25-39.
- [12] C. Kern and M. R. Greenstreet, "Formal verification in hardware design: a survey," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 4, no. 2, pp. 123-193, 1999.
- [13] V. Mascardi, M. Martelli, and L. Sterling, "Logic-based specification languages for intelligent

software agents," *Theory and Practice of Logic Programming*, vol. 4, no. 4, pp. 429-494, 2004.

[14] D. Bjørner and M. C. Henson, *Logics of specification languages*. Springer Science & Business Media, 2007.

[15] M. Krichen, "A survey on formal verification and validation techniques for internet of things," *Applied Sciences*, vol. 13, no. 14, p. 8122, 2023.

[16] B. Andersson, D. de Niz, W. Vance, J. Ross, M. Wotell, and T. Bui, "Methodology of Combining Empirical Stress Testing and Formal-Methods Based Schedulability Analysis for Real-Time Multicore Software," in *2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC)*, 2023: IEEE, pp. 1-10.

[17] J. C. Knight, "Safety critical systems: challenges and directions," in *Proceedings of the 24th international conference on software engineering*, 2002, pp. 547-550.

[18] Z. Liu, X. Zhang, Y. Sun, and Y. Zhou, "Advanced controls on energy reliability, flexibility and occupant-centric control for smart and energy-efficient buildings," *Energy and Buildings*, vol. 297, p. 113436, 2023/10/15/ 2023, doi: <https://doi.org/10.1016/j.enbuild.2023.113436>.

[19] Z. Liu *et al.*, "Incentive initiatives on energy-efficient renovation of existing buildings towards carbon-neutral blueprints in China: Advancements, challenges and prospects," *Energy and Buildings*, vol. 296, p. 113343, 2023/10/01/ 2023, doi: <https://doi.org/10.1016/j.enbuild.2023.113343>.

[20] D. Rodríguez-Gracia, M. d. I. M. Capobianco-Uriarte, E. Terán-Yépez, J. A. Piedra-Fernández, L. Iribarne, and R. Ayala, "Review of artificial intelligence techniques in green/smart buildings," *Sustainable Computing: Informatics and Systems*, vol. 38, p. 100861, 2023/04/01/ 2023, doi: <https://doi.org/10.1016/j.suscom.2023.100861>.

[21] A. A. Khan, M. A. Khan, K. Leung, X. Huang, M. Luo, and A. Usmani, "A review of critical fire event library for buildings and safety framework for smart firefighting," *International Journal of Disaster Risk Reduction*, vol. 83, p. 103412, 2022/12/01/ 2022, doi: <https://doi.org/10.1016/j.ijdr.2022.103412>.

[22] Z. Liu *et al.*, "A review of data-driven smart building-integrated photovoltaic systems: Challenges and objectives," *Energy*, vol. 263, p. 126082, 2023/01/15/ 2023, doi: <https://doi.org/10.1016/j.energy.2022.126082>.

[23] H. Xia, Z. Liu, M. Efremochkina, X. Liu, and C. Lin, "Study on city digital twin technologies for sustainable smart city design: A review and bibliometric analysis of geographic information system and building information modeling integration," *Sustainable Cities and Society*, vol. 84, p. 104009, 2022/09/01/ 2022, doi: <https://doi.org/10.1016/j.scs.2022.104009>.

[24] G. Pinto, Z. Wang, A. Roy, T. Hong, and A. Capozzoli, "Transfer learning for smart buildings: A critical review of algorithms, applications, and future

perspectives," *Advances in Applied Energy*, vol. 5, p. 100084, 2022/02/01/ 2022, doi: <https://doi.org/10.1016/j.adapen.2022.100084>.

[25] R. Dwivedi, D. Mehrotra, and S. Chandra, "Potential of Internet of Medical Things (IoMT) applications in building a smart healthcare system: A systematic review," *Journal of Oral Biology and Craniofacial Research*, vol. 12, no. 2, pp. 302-318, 2022/03/01/ 2022, doi: <https://doi.org/10.1016/j.jobocr.2021.11.010>.

[26] G. F. Huseien and K. W. Shah, "A review on 5G technology for smart energy management and smart buildings in Singapore," *Energy and AI*, vol. 7, p. 100116, 2022/01/01/ 2022, doi: <https://doi.org/10.1016/j.egyai.2021.100116>.

[27] J. Aguilar, A. Garces-Jimenez, M. D. R-Moreno, and R. García, "A systematic literature review on the use of artificial intelligence in energy self-management in smart buildings," *Renewable and Sustainable Energy Reviews*, vol. 151, p. 111530, 2021/11/01/ 2021, doi: <https://doi.org/10.1016/j.rser.2021.111530>.

[28] A. Malagnino, T. Montanaro, M. Lazoi, I. Sergi, A. Corallo, and L. Patrono, "Building Information Modeling and Internet of Things integration for smart and sustainable environments: A review," *Journal of Cleaner Production*, vol. 312, p. 127716, 2021/08/20/ 2021, doi: <https://doi.org/10.1016/j.jclepro.2021.127716>.

[29] H. Stopps, B. Huchuk, M. F. Touchie, and W. O'Brien, "Is anyone home? A critical review of occupant-centric smart HVAC controls implementations in residential buildings," *Building and Environment*, vol. 187, p. 107369, 2021/01/01/ 2021, doi: <https://doi.org/10.1016/j.buildenv.2020.107369>.

[30] J. Al Dakheel, C. Del Pero, N. Aste, and F. Leonforte, "Smart buildings features and key performance indicators: A review," *Sustainable Cities and Society*, vol. 61, p. 102328, 2020/10/01/ 2020, doi: <https://doi.org/10.1016/j.scs.2020.102328>.

[31] B. Dong, V. Prakash, F. Feng, and Z. O'Neill, "A review of smart building sensing system for better indoor environment control," *Energy and Buildings*, vol. 199, pp. 29-46, 2019/09/15/ 2019, doi: <https://doi.org/10.1016/j.enbuild.2019.06.025>.

[32] M. Jia, A. Komeily, Y. Wang, and R. S. Srinivasan, "Adopting Internet of Things for the development of smart buildings: A review of enabling technologies and applications," *Automation in Construction*, vol. 101, pp. 111-126, 2019/05/01/ 2019, doi: <https://doi.org/10.1016/j.autcon.2019.01.023>.

[33] L. Lamport, "Specifying systems: the TLA+ language and tools for hardware and software engineers," 2002.

[34] A. Rehman, S. Latif, and N. A. Zafar, "Formal modeling of smart office using activity diagram and non deterministic finite automata," in *2019 International Conference on Information Science and Communication Technology (ICISCT)*, 2019: IEEE, pp. 1-5.

- [35] S. Latif, A. Rehman, and N. A. Zafar, "Modeling of sewerage system linking UML, automata and TLA+," in *2018 international conference on computing, electronic and electrical engineering (ICE Cube)*, 2018: IEEE, pp.1-6.
- [36] R. Adner and D. Levinthal, "Demand heterogeneity and technology evolution: implications for product and process innovation," *Management science*, vol. 47, no. 5, pp. 611-628, 2001.
- [37] F. K. Santoso and N. C. Vun, "Securing IoT for smart home system," in *2015 international symposium on consumer electronics (ISCE)*, 2015: IEEE, pp. 1-2.
- [38] Z. Yu, Y. Liang, B. Xu, Y. Yang, and B. Guo, "Towards a smart campus with mobile social networking," in *2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing*, 2011: IEEE, pp. 162-169.
- [39] K. Furdik, G. Lukac, T. Sabol, and P. Kostelnik, "The network architecture designed for an adaptable IoT-based smart office solution," *International Journal of Computer Networks and Communications Security*, vol. 1, no. 6, pp. 216-224, 2013.
- [40] Y. Geng and C. G. Cassandras, "New "smart parking" system based on resource allocation and reservations," *IEEE Transactions on intelligent transportation systems*, vol. 14, no. 3, pp. 1129-1139, 2013.
- [41] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things journal*, vol. 1, no. 1, pp. 22-32, 2014.
- [42] L. Pocero, D. Amaxilatis, G. Mylonas, and I. Chatzigiannakis, "Open source IoT meter devices for smart and energy-efficient school buildings," *HardwareX*, vol. 1, pp. 54-67, 2017.
- [43] D. Amaxilatis, I. Chatzigiannakis, and G. Mylonas, "Design and Implementation of a Platform for Smart Connected School Buildings," in *Aml (Workshops/ Posters)*, 2015.
- [44] Y. Qu, H. Wang, S.-M. Lun, H.-D. Chiang, and T. Wang, "Design and implementation of a Web-based Energy Management Application for smart buildings," in *2013 IEEE Electrical Power & Energy Conference*, 2013: IEEE, pp. 1-6.
- [45] M. Brogan and A. Galata, "The Vervyschool project: Valuable energy for a smart school-intelligent ISO 50001 energy management decision making in school buildings In Proceedings of the Special Tracks and Workshops at the 11th International Conference on Artificial Intelligence Applications and Innovations (AIAI 2015), Bayonne, France, September 14-17, 2015," ed, 2015.
- [46] B. Hirsch and J. W. Ng, "Education beyond the cloud: Anytime-anywhere learning in a smart campus environment," in *2011 international conference for internet technology and secured transactions*, 2011: IEEE, pp. 718-723.
- [47] R. Huang, J. Zhang, Y. Hu, and J. Yang, "Smart campus: The developing trends of digital campus," *Open education research*, vol. 4, no. 004, 2012.
- [48] M. Veeramanickam and M. Mohanapriya, "lot enabled futururus smart campus with effective e-learning: i-campus," *GSTF journal of Engineering Technology (JET)*, vol. 3, no. 4, pp. 8-87, 2016.
- [49] G. Smith, *The Object-Z specification language*. Springer Science & Business Media, 2012.
- [50] P. G. Larsen and W. Pawlowski, "The formal semantics of ISO VDM-SL," *Computer standards & interfaces*, vol. 17, no. 5-6, pp. 585-601, 1995.
- [51] D. Jackson, "Alloy: a lightweight object modelling notation," *ACM Transactions on software engineering and methodology (TOSEM)*, vol. 11, no. 2, pp. 256-290, 2002.
- [52] S. Latif, A. Rehman, and N. A. Zafar, "NFA based formal modeling of smart parking system using TLA+," in *2019 international conference on information science and communication technology (ICISCT)*, 2019: IEEE, pp. 1-6.
- [53] T. De Champs, M. Ouenzar, B. Abdulrazak, M. Frappier, H. Pigot, and B. Fraikin, "Pervasive safety application with model checking in smart houses: The INOVUS intelligent oven," in *2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 2011: IEEE, pp. 630-635.
- [54] E. Cohen et al., "VCC: A practical system for verifying concurrent C," in *Theorem Proving in Higher Order Logics: 22nd International Conference, TPHOLs 2009, Munich, Germany, August 17-20, 2009. Proceedings 22, 2009: Springer*, pp. 23-42.
- [55] K. Lano, *The B language and method: a guide to practical formal development*. Springer Science & Business Media, 2012.
- [56] J.-R. Abrial, M. Butler, S. Hallerstede, T. S. Hoang, F. Mehta, and L. Voisin, "Rodin: an open toolset for modelling and reasoning in Event-B," *International journal on software tools for technology transfer*, vol. 12, pp. 447-466, 2010.
- [57] H. Koç, A. M. Erdoğan, Y. Barjakly, and S. Peker, "UML diagrams in software engineering research: a systematic literature review," in *Proceedings*, 2021, vol. 74, no. 1: MDPI, p. 13.
- [58] E. B. Purnama, N. Azizah, M. Nuraminudin, and M. M. Dewi, "Analysis and Design of Queue Service Information System Integrated with WhatsApp using UML Method," *International Journal of Computer and Information System (IJCIS)*, vol. 5, no. 1, pp. 1-7, 2024.
- [59] E. M. Clarke, "Model checking," in *Foundations of Software Technology and Theoretical Computer Science: 17th Conference Kharagpur, India, December 18-20, 1997 Proceedings 17, 1997: Springer*, pp.54-56.
- [60] Clarke, E. M, and Emerson, E. A. (1981, May). Design and synthesis of synchronization skeletons using branching time temporal logic. In *Workshop on logic of programs* (pp. 52-71). Berlin, Heidelberg: Springer Berlin Heidelberg.

- [61] C. Baier and J.-P. Katoen, *Principles of model checking*. MIT press, 2008.
- [62] R. Jhala and R. Majumdar, "Software model checking," *ACM Computing Surveys (CSUR)*, vol. 41, no. 4, pp. 1-54, 2009.
- [63] B. Bérard et al., *Systems and software verification: model-checking techniques and tools*. Springer Science & Business Media, 2013.
- [64] I. Bouziane, H. Belmokadem, and M. Moussaoui, "A Review of Formal Security Verification of Common Internet of Things (IoT) Communication Protocols," in *2023 7th IEEE Congress on Information Science and Technology (CiSt)*, 2023: IEEE, pp. 334-341.
- [65] M. Shafiq, Z. Gu, O. Cheikhrouhou, W. Alhakami, and H. Hamam, "The rise of "Internet of Things": review and open research issues related to detection and prevention of IoT-based security attacks," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1-12, 2022.
- [66] K. Hofer-Schmitz and B. Stojanović, "Towards formal verification of IoT protocols: A Review," *Computer Networks*, vol. 174, p. 107233, 2020.
- [67] A. Souri and M. Norouzi, "A state-of-the-art survey on formal verification of the internet of things applications," *Journal of Service Science Research*, vol. 11, no. 1, pp. 47-67, 2019.
- [68] J. Sun and Y. Liu, "Design of 360° Dead-Angle-Free Smart Desk Lamp based on Visual Tracking," *HighTech and Innovation Journal*, vol. 4, no. 4, pp. 761-767, 2023.
- [69] S. Amhaimedi, S. Naimi, and S. Alsallami, "Assessment of a decision-making model for monitoring the success of a project for smart buildings," 2023.
- [70] M. S. Sayeed, H. Abdulrahim, S. F. A. Razak, U. A. Bukar, and S. Yogarayan, "IoT Raspberry Pi Based Smart Parking System with Weighted K-Nearest Neighbours Approach," *Civil Engineering Journal*, vol. 9, no. 8, pp. 1991-2011, 2023.
- [71] C. Newcombe, T. Rath, F. Zhang, B. Munteanu, M. Brooker, and M. Deardeuff, "How Amazon web services uses formal methods," *Communications of the ACM*, vol. 58, no. 4, pp. 66-73, 2015.
- [72] M. Nielsen, K. Havelund, K. R. Wagner, and C. George, "The RAISE language, method and tools," *Formal aspects of computing*, vol. 1, pp. 85-114, 1989.
- [73] N. Plat and P. G. Larsen, "An overview of the ISO/VDM-SL standard," *ACM Sigplan Notices*, vol. 27, no. 8, pp. 76-82, 1992.

AUTHOR BIOGRAPHY



SHAHBAZ AHMAD received Master's Degree in Computer Science from Islamia University, Bahawalpur, Pakistan in 2013 and an M.Phil degree in Computer Science from NCBA& E (National College of Business Administration & Economics, Lahore), Pakistan in 2018. He is currently pursuing the Ph.D. degree at Islamia University, Bahawalpur, Punjab, Pakistan. His research interests include IoT, Information Security, Network Security, Wireless Network, Machine Learning, Artificial Intelligence, Data Science, and Formal Methods.



ADEEL AHMAD received the Master's degree in computer science from The Islamia University of Bahawalpur, Pakistan, and the M.S. degree in computer sciences from the Virtual University of Pakistan. He has also received the Ph.D. degree from the Islamia University of Bahawalpur. His main research interests include edge computing, IoT systems, energy efficiency, fuzzy logic, wireless networks, and blockchain.



NADEEM AKHTAR received the Ph.D. degree (Hons.) from the Laboratory VALORIA of Computer Science, University of South Brittany (UBS), France, and the master's degree specialization in information systems from the Archite Institut Universitaire Professionnalise Vannes, France. He is currently working as an Assistant Professor with the Faculty of Computing, The Islamia University of Bahawalpur (IUB), Pakistan. He was a recipient of several awards, scholarships, and research grants, i.e., a study in France, 2004 French Embassy scholarship for Master studies, HEC, France, as a teaching assistant for ENSIBS = million, research award in 2014 with The Directorate of Research and Development, The Islamia University of Bahawalpur.



SAIMA ABDULLAH received the Ph.D. degree from the Department of Computer Science and Electronic Engineering, University of Essex, U.K. She is currently an Assistant Professor with the Department of Computer Science and Information Technology, The Islamia University of Bahawalpur, Pakistan. She is a member of the Multimedia Research Group, DCS, where she has been involved in efficient and secure communication of multimedia data over future generation network technologies. Her main research interests include wireless networks and communications, future internet technology, and network performance analysis. She has authored around ten articles in the above research areas. She serves as a reviewer for international journals.