Indonesian Journal of Electrical Engineering and Computer Science
Vol. 41, No. 1, January 2026, pp. 233~249
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v41.i1.pp233-249

Tool support for LoRaWAN development: a comparative perspective on simulation and emulation

Ntshabele Koketso, Bassey Isong
Department of Computer Science, Faculty of Natural and Agricultural Sciences, North-West University, Mafikeng, South Africa

Article Info

Article history:

Received Jul 30, 2025
Revised Nov 16, 2025
Accepted Dec 13, 2025

Keywords:

Emulation tools
Internet of things
Long range wireless area network security
Simulation tools

ABSTRACT

This paper explores the use of various long range wireless area network (LoRaWAN) simulation and emulation tools when designing and evaluating IoT networks. Simulation tools are often popular with researchers because they are less costly and can easily simulate large-scale networks, allowing for easy and faster tests of the scalability of various protocols and behaviors. However, they often lack the unpredictable nature of real deployments. Emulation and cloud-based tools fill this gap, but with their flexibility they provide a more realistic approximation of real-world performance and allow easier interfacing with actual network hardware infrastructure, although they generally incur a higher cost which is often controlled by technical skill level use.

Corresponding Author:

Koketso Ntshabele
Department of Computer Science, Faculty of Natural and Agricultural Sciences
North-West University
Mafikeng, South Africa
Email: koketso.ntshabele@nwu.ac.za

1. INTRODUCTION

The internet of things (IoT) is made up of constrained physical devices that can create and share data with no direct human intervention. The IoT normally consists of sensing, communication, processing, and user interaction layers [1], [2]. Among the numerous IoT communication technologies, long range wireless area network (LoRaWAN) has emerged as one of the most popular, attracting wide interest in applications that call for long-range connectivity coupled with low power consumption, as stated in [1], [3]. LoRaWAN was developed by Semtech. It makes use of unlicensed frequency bands combined with chirp spread spectrum (CSS) modulation. The result is the ability to communicate securely with devices that have longer battery life, as well as efficient communication, which leads to longer battery life, as discussed in [1], [2], [4]. The star-of-stars topology lets endpoints send data to a central network server via gateways, thus allowing the monitoring, control, and scalability of networks remotely [1], [2], [5]–[8].

LoRaWAN offers a balance between deployment security, flexibility, and efficiency through device activation by either over-the-air-activation (OTAA) or activation-by-personalization (ABP) [5], [6], [8]. To align with different power and latency requirements, LoRaWAN offers authentication, integrity, and encryption, techniques where devices are implemented in one of three LoRaWAN classes: A, B, or C [2], [5], [6], [8]. However, though LoRaWAN offers security, the network is still susceptible to data threats and attacks such as eavesdropping, data tampering, illegal access [9]–[11].

Simulators and emulators are critical elements in LoRaWAN structure and functional design and optimization. They allow simulating network topology, signal transmission, data flow, energy consumption, and compatibility. They are useful in identifying bottlenecks, protocol validation, and eliminating

susceptibility to attack [2], [6], [12]–[14]. One normally appreciates open-source platforms since they are flexible, affordable, do not rely on vendors, and undergo rigorous tests to ensure that security parameters are met [15], [16]. Notwithstanding the current increasing adoption of LoRaWAN technology, researchers are still challenged to identify appropriate simulators and emulators due to the numerous platforms and lack of guidance [6], [7], [11]. The general LoRaWAN architecture, including end devices, gateways, a network server, and applications, is shown in Figure 1. It highlights how secure socket layer (SSL) is used to facilitate secure communication flows at both the transmission control protocol/internet protocol (TCP/IP) and long range (LoRa) levels. The performance, scalability, usability, flexibility, device class support, licensing, and integrations offered by a number of well-known LoRaWAN tools are examined in this paper in order to close this gap. The objective is to encourage further work on the development of the LoRaWAN protocol and application design while offering helpful advice on tool selection. The remainder of the paper is structured as follows: Section 2 provides literature review, section 3 is the methodology. Section 4 offers critical results and discussion, while section v concludes the study.

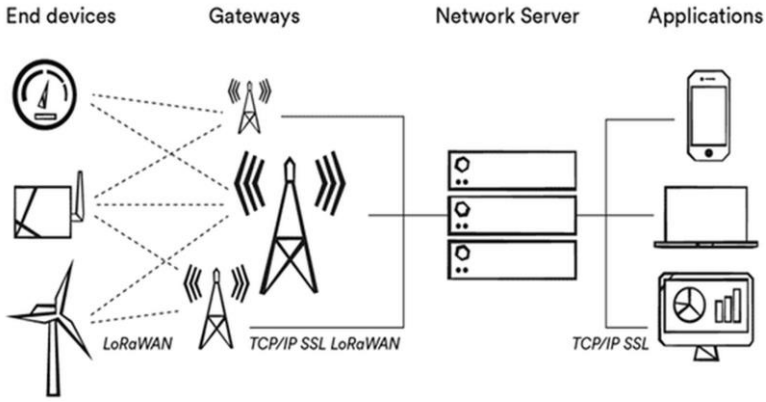

Figure 1. LoRaWAN architecture

2. LITERATURE REVIEW

The need to create models that are balanced in terms of accuracy and practical application has been deepening regarding development technologies within the LoRaWAN environment, as is evident from the current literature. To study the outcomes in terms of IoT applications, simulations as well as emulator models have been researched.

2.1. Related works

Several studies have evaluated LoRaWAN simulators, focusing primarily on performance features within specific frameworks. For instance, [17] surveyed NS-3 modules to assist researchers in choosing suitable LPWAN components, while [1] and [2] offered broader analyses emphasizing design requirements, experiment types, and performance constraints. Idris *et al.* [18] further examined key metrics such as CPU load, memory usage, collisions, and delivery rates across three common simulators, though broader evaluations remain necessary. Other works, such as Almuhaya *et al.* [19], compared LPWAN technologies, highlighting LoRa's strengths, yet their assessment lacked detailed comparisons of simulation environments. Similarly, Marini *et al.* [5] analyzed LoRaWANSim using MATLAB, calling out simulation oversimplifications but did not benchmark it against alternatives, underscoring the need for comparative studies across simulation layers.

Most existing literature, including [1], focuses narrowly on performance and scalability. Interoperability and security assessments are sparse, with recent studies [4]-[6], [8] addressing efficiency but lacking depth on energy-aware security frameworks. Encryption and authentication have been explored, yet cross-tool evaluations remain limited. Another challenge is the inconsistency in evaluation methods. Without standardized benchmarking, comparisons lose reliability. Hardware-in-the-loop (HIL) models are emerging to bridge this gap by injecting real-world conditions, such as interference and congestion, into simulation contexts. However, theoretical testing still dominates, missing environmental variables. Introducing machine learning approaches may improve dynamic threat detection and optimize tool selection. This paper contributes by addressing current gaps through a comparative analysis of modern LoRaWAN emulators and simulators, guiding selection based on specific application needs and security parameters.

This paper addresses these gaps by providing a comparative analysis of modern LoRaWAN simulators and emulators. The goal is not to declare a single best platform, but to help researchers and developers choose the tools that best fit their project goals, especially when considering security, scalability, and specific application needs.

2.2. LoRaWAN state-of-art tools

This section outlines both simulation tools, which focus on modeling network performance, and emulators, which are used for testing more realistic system interactions. The summaries in Tables 1–6 highlight key characteristics, including performance metrics, infrastructure, device support, and available security mechanisms.

2.2.1. LoRAWAN simulation tools

Some of the existing simulation tools for LoRaWAN network creation and testing are discussed in this subsection. Tables 1 and 2 summarize the tools based on usability, scalability, licensing, support, device class compatibility, and integration. The aim is to equip researchers with comparative insights to support informed tool selection.

LoRaSim, an MIT-licensed open-standard simulator, facilitates performance analysis of LoRaWAN networks across varied deployment scenarios. It supports customizable traffic patterns, network topologies, and radio configurations, and offers GUI-based interaction. Widely adopted in research, it enables protocol testing and attack simulations, promoting scalable, cross-integrated usage [1], [2], [6]. Similarly, NS-3, governed by the GNU GPLv2, is a discrete-event simulator extensively employed in research and education. It allows evaluation of diverse network protocols, including TCP congestion algorithms, SDN architectures, and routing models, and supports scalable deployments. Integrated with robust security modules, NS-3 enables testing of attack mitigation strategies across LoRaWAN configurations [1], [2], [20].

Another versatile option is FloRa, a flexible open-source simulator, is designed for ease of use and interoperability with other IoT platforms. Though not technology-specific and lacking defined security protocols, it supports all LoRaWAN device classes and allows scalable network setups through a growing developer and vendor community [1], [21]–[23]. Also, LoRaEnergySim, distributed under the MIT license, specialises in modelling energy consumption for LoRaWAN devices. It accommodates small to large networks, supports all device classes, and integrates with external tools, though it omits encryption and authentication features [2], [24]. LoRaFree equally provides a browser-based interface with configurable network management. It supports device authentication and encryption, all LoRaWAN classes, and interoperability with third-party platforms via APIs like WebSockets, HTTP, and MQTT. Licensed under Apache 2.0, it enables unrestricted modification and distribution, although support resources are limited to GitHub and community forums [4].

Similarly, CupCarbon, a Java-based open-source simulator, offers GUI-driven modelling of radio propagation, energy use, and packet delivery across diverse devices. Capable of simulating thousands of nodes, including ESP32, Android, and IoTNode variants, it integrates with platforms like MATLAB, NS-3, and OMNeT++. Educational use is permitted under the GNU General Public License, with extensive documentation and tutorials available [13], [19], [25]. In the same vein, SimpleIoTSimulator supports a broad array of network types and devices, including LoRa sensors and MQTT clients, and features a wizard interface with message replay capabilities. It can simulate extensive networks and analyse key security mechanisms such as encryption, authentication, and key management, backed by documentation, demos, and online resources [26]–[28]. In the same note, Mbed Simulator is an open-source solution supporting complex network configurations, including LoRa sensors and gateways, across both online and offline versions of Mbed OS 5. Distributed under the Apache 2.0 License, users are required to cite its use in publications. Though it does not explicitly detail built-in security features, integration with platforms like MATLAB and support for all device classes make it versatile. Numerous studies have nevertheless employed Mbed Simulator to assess key security mechanisms, including encryption, authentication, and key management [19], [29].

OMNeT++, governed by the GNU General Public License, is widely used in academic research for non-commercial purposes. It offers a GUI and command-line interface for model design, simulation execution, and results analysis. Supporting C++ node programming and capable of handling thousands of nodes, OMNeT++ accommodates all LoRaWAN device classes and integrates with external platforms. While its documentation does not highlight LoRaWAN-specific security modules, it has been extensively applied in studies analyzing encryption protocols and authentication frameworks [21]-[23]. In a similar design, another important tool is the Network Simulator. It is a freely available open-source tool equipped with a user-friendly interface and compatible with varying topologies, channel models, and modulation schemes. Network Simulator supports all three device classes of LoRaWAN and even provides APIs for use by third-party applications. Although highly compatible, its documentation does not provide much information about

security features. Nonetheless, it has been used in some studies to analyzed how secure the protocol is when under attack [1], [5], [24]. Correspondingly, LoRaSimu is an open-source, web-based simulator that complies with LoRa Alliance Terms of Use. It can support all device classes, reconfigurable topologies, and all the payload formats and permits network model creation and modification. Its lack of built-in authentication and encryption features [5], and support for integration into third-party platforms renders it less than fully appropriate to be used for simulating secure networks.

Table 1. LoRaWAN simulation tools summary

Tool	Usability	Scalability	Flexibility	Licensing	Support	Device class	Integration
LoRaSim [1], [2], [5]	Simple, user-friendly interface	Small to large networks	Configurable topologies, modulation schemes	MIT (Open-source)	GitHub community, documentation	A, B, C	OMNeT++, R, MATLAB
NS-3 [17]	Requires programming proficiency	Small to large networks	Advanced customisation, protocol flexibility	GNU GPLv2	Active community, mailing lists	A, B, C	NS-3, MATLAB, OMNeT++, Python
FloRa [21]-[23]	Intuitive interface via OMNeT++ framework	Small to large networks	Dynamic ADR support, LoRa-specific settings	MIT (Open standard)	Aalto University & developer forums	A, B, C	SimuLTE, Veins, IoT platforms
LoRaEnergySim [2], [24].	Straightforward interface	Small to large networks	Energy-focused, customizable parameters	MIT (Open-source)	Community forums	A, B, C	Supports external tools
LoRaFree [4]	Web-based, user-friendly GUI	Small- to medium-scale	Flexible payload and device configurations	Apache 2.0	GitHub-based support	A, B, C	WebSockets, HTTP, MQTT APIs
CupCarbon [13], [25]	Intuitive GUI with OSM support	Small to large networks	Node scripting (Python/SenScript), MQTT	GNU General Public License	Tutorials, GitHub, online docs	A, B, C	NS-3, MATLAB, OMNeT++, FloRa
SimpleToSimulator [28]	Wizard-style GUI	Small to large networks	Customizable payload and gateway simulations	Licensed with citation required	Product sheets, demo videos	A, B, C	NS-3, CupCarbon, MATLAB, OMNeT++
Mbed Simulator [29]	Web-based interface + offline OS 5	Small to large networks	C++ support, MQTT-enabled testing	Apache 2.0	GitHub, blogs, video tutorials	A, B, C	NS-3, MATLAB, CupCarbon, FloRa

Another useful tool providing an easy-to-use interface for developing and experimenting with large-scale network models is the LoRaWAN Simulator. Along with device customisation and payload formatting options, it provides support for basic protocols like ABP, OTAA, and ADR. With encryption and key management options available, security becomes a priority. Its academic as well as commercial usability is attested to by its ease of interfacing with other external packages and comprehensive, well-documented documentation [1], [5], [12]. An open-source LoRaWAN network server licensed under Apache 2.0, ChirpStack boasts a very excellent record for scalability and flexibility in commercial and research deployments. It supports secure communication protocols, facilitates capacity for thousands of devices and gateways, and has a modern web-based interface. Long-term viability and ongoing development are guaranteed by the ongoing contributions of an engaged developer community [9], [35], [36]. The things network (TTN), a free, GPL-licensed initiative, offers an open-source platform, globally accessible and designed for cost-effective LoRaWAN solutions. It boasts reliable data storage, solid device management, and secure application integration. Its universality of use among academic and businesspeople is an indication of its worth and affordability [6], [12], [30].

Other comparable open-source infrastructure appropriate for low-cost LoRaWAN systems is that of TTN under the GPL license. It is integrated with secure apps, efficient in the management of devices, and stores data securely. It is widely utilised in academic as well as corporate circles, which is a pointer to its efficacy and low cost [6], [12], [30]. It supports a large variety of devices and gateways; however, ease of use remains a challenge for users, and the costs involved can be a dampener. Its robust customer support and suitability for smart city and agricultural applications make it a suitable choice for focused deployments [6], [12], [30]. In addition, LoRaWAN Test Suites provide standardised validation environments across all device classes, adaptive rate configurations, and spreading factors. They do not offer API integration or support

from community-driven forums. However, they fully implement encryption, authentication, and authorisation protocols, making them suitable for secure network certification and compliance assessment [34].

Table 2. LoRaWAN simulation tools summary

Tool	Usability	Scalability	Flexibility	Licensing	Support	Device class	Integration
OMNeT++ [21-[23]	GUI/CLI combo for detailed simulations	Small to large networks	Visual modelling, customizable C++ scripts	GNU General Public License	Manuals, tutorials, GitHub	A, B, C	NS-3, MATLAB, FloRa, CupCarbon
LoRaWAN Network Simulator [1], [2], [6]	GUI-based modelling	Small to large networks	Channel settings, modulation support	Open source via LoRa Alliance	Forums, documentation	A, B, C	APIs for simulator interoperability
LoRaSimu [5]	Browser-based modelling and analysis	Multi-node scenarios	Payload and network parameter configuration	Open source via LoRa Alliance	Contact form, feedback channels	A, B, C	NS-3, MATLAB, CupCarbon, FloRa
LoRaWAN Simulator [1], [5]	GUI-enabled modelling suite	Multi-node networks	Custom device payload and configuration	Open source via LoRa Alliance	Contact form, feedback channels	A, B, C	NS-3, MATLAB, CupCarbon, FloRa
TTN [6], [12], [30]	Web-based, easy-to-manage, and easy-to-test networks	Large-scale LoRaWAN deployments	Open infrastructure supports secure communication and app integration	GNU GPL	Community docs, forums, developer support	A, B, C	NS-3, MATLAB, FloRa, CupCarbon, APIs
Loriot [10], [20], [31]	GUI with remote access, ADR, geolocation, cloud/on-prem deployment	Highly scalable, enterprise-grade	Supports varied deployment models and IoT applications	Proprietary	Email, phone, docs, knowledge base	A, B, C	NS-3, OMNeT++, APIs, enterprise platforms
Lorix One [32], [33]	Cloud-based device/gateway management	Scalable across multi-device setups	ADR, geolocation, remote monitoring; integrates with other platforms	Proprietary	Email, phone, docs (may be limited)	A, B, C	APIs, IoT platforms, interoperable with TTN, Loriot
LoRaWAN Test Suites [34]	CLI and GUI for compliance and certification testing	Cross-regional and cross-class support	Tests Class A/B/C, ADR, spreading factors	Free toolset	User guide; no dedicated support portal	A, B, C	USB or RF interface only; no external API integration

Tables 3 and 4 offers a complete analysis of various simulation solutions that exist for the LoRaWAN technology with the key strengths and intrinsic weaknesses to be understood with regard to each one's performance requirements, infrastructural requirements, device simulation capabilities, services offered and offered levels of resilience. Of these simulation solutions, the likes of LoRaSIM, NS-3, and FloRaSim are seen to be doing extensive work regarding simulating networking efficiencies and energy efficiencies with the help of packet routing measures that entail delivery rates, delays, and transfer capacities. But then again, the areas that most simulation solutions are deficient in include those that concern either scant simulations with regard to cryptographic models designed to be more accurate in terms of encrypting data packets that are beyond either the simulation solution's intrinsic capacities or that are entrusted to outside libraries such as the one that supports TLS. The minimalist simulation solution, either that of 'LoRaEnergySim' or 'LoRaFree', offers relative emphasis on either energy efficiency measures and 'web user friendlyness', respectively.

Similarly, there are other emulators such as OMNeT++, LoRaWan network simulator that are greatly customizable to enable the researcher to test with a realistic topology; these are also modular and can be applied in the industry as well as in research. Other web-based solutions, such as LoRaSimu and LoRaWAN Simulator, increase the ease of use and flexibility for scenarios but, usually cannot provide strong encryption layers for keeping messages secure. In contrast, the ChirpStack and Loriot tools had the best performance for a cloud-grade and enterprise-level deployment since their architecture comprises TLS/SSL encryption, OAuth2 authentication, and role-based access control, making them more security-aware. Lorix One and LoRaWAN Test Suites extend the ecosystem toward device validation, compliance, and gateway management, but some of them rely on external platforms-for example, TTN or Loriot-for the full enforcement of security. As such, Table 4 underlines a continuous movement from academic simulators to enterprise-grade frameworks, even though it highlights the need for standardized, secure, and interoperable simulation environments in LoRaWAN research and deployment.

Table 3. Summarized limitations of LoRaWAN simulation tools

Tools	Performance	Metrics	Infrastructure	Device Support	Services	Security
LoRaSIM [1], [2]	High simulation fidelity for network/device behaviours	Packet delivery, latency, energy, topology changes	Desktop GUI; supports OMNeT++, MATLAB, R	A, B, C	Channel modelling, modulation testing	Customizable simulations; no built-in encryption/auth
NS-3 [17]	Protocol-rich, advanced stack simulation	BER, packet loss, throughput, energy efficiency	Linux CLI/scripting (C++, Python)	A, B, C	Routing protocols, real-time simulation	Manual protocol modelling; external libraries for TLS
FloRa [21]-[23]	Energy-efficiency and ADR-focused	Energy profiles, latency, throughput, ADR metrics	OMNeT++ with INET framework	A, B, C	IoT-focused: SimuLTE, Veins integrations	Basic LoRaWAN security validation
LoRaEnergySim [2], [24].	Lightweight energy modelling	Power use, transmission delay, node lifetime	Desktop, minimal hardware	A, B, C	Payload customisation, frequency/data rate tuning	No complete protocol simulation; energy focus only
LoRaFree [4]	Real-time browser-based simulations	Packet loss, latency, node distance	Web with cloud integration	A, B, C	Network visualisation, profiling	Basic only; lacks encryption/authentication
CupCarbon [13], [25]	Urban-scale simulations with mobile sensors	RSSI, packet rate, mobility, location modelling	Java GUI with OpenStreetMap	A, B, C	SenScript, IoT modelling, real-device emulation	MQTT-based encryption lacks secure channel modelling
SimpleToSimulator [28]	Gateway-payload interaction modelling	Success rates, latency, gateway use	Wizard-style GUI; paid tool	A, B, C	Auto scenarios, node interaction	Secured features vary by plan; encryption docs limited
Mbed Simulator [29]	Embedded/cloud simulation for LoRa nodes	Uptime, MQTT delivery, battery impact	Online/offline; supports Mbed OS 5	A, B, C	MQTT testing, cloud sync via C++	TLS encryption, Mbed trust anchors supported

Table 4. Summarised limitations of LoRaWAN simulation tools

Tools	Performance	Metrics	Infrastructure	Device Support	Services	Security
OMNeT++ [21]-[23]	Modular platform with deep customisation	Node interaction, success ratio, delay	Desktop app; INET, Veins, SimuLTE extension	A, B, C	Modular simulations, visual network modelling	Varies by module; encryption logic customizable
LoRaWAN Network Simulator [1], [2], [6]	Full-stack LoRaWAN modelling	Node coverage, frame reception, topology scaling	Desktop/server; LoRa Alliance-based	A, B, C	Topology testing, protocol support	Basic security modelling; citation required for extended use
LoRaSimu [5]	High-fidelity packet and network simulation	Gateway reception, signal strength, loss	Web-based simulation engine	A, B, C	Gateway, node, MAC layer simulation	Protocol-based; lacks secure communication layers
LoRaWAN Simulator [1], [5]	Interactive topology and node testing	Frame repetition, delivery, node range	Web platform (LoRa Alliance compliant)	A, B, C	Scenario editing, gateway simulation	Security behaviour testing only; lacks full encryption
ChirpStack [9], [35], [36]	Cloud-grade performance with network services	Throughput, gateway load, connection reliability	Distributed server with web admin UI	A, B, C	MQTT integration, device provisioning	TLS/SSL transport, OAuth2, token auth, role-based access control
Loriot [10], [20], [31]	Enterprise-grade for hybrid models	Uptime, delivery rates, ADR success, geolocation	Public cloud and on-prem options	A, B, C	Monitoring, gateway control, custom API support	Role-based access, encrypted data streams, secure API management
Lorix One [32], [33]	Flexible and robust under cloud-managed operations	Packet throughput, gateway connectivity, uptime	Cloud UI via Lorix or partner integrations	A, B, C	Network integration, remote control, gateway provisioning	LoRaWAN-based encryption; full security via TTN or Loriot integrations
LoRaWAN Test Suites [34]	Precise tool for certification and protocol testing	RSSI, PER, frequency offset, power levels	USB or RF interface; CLI/GUI test suite	A, B, C	Compliance and benchmarking validation	Authentication/encryption test cases; lacks enforcement of secure transmission

2.2.2. LoRaWAN emulator tools

This subsection discusses some of the key emulator platforms for LoRaWAN deployments as shown in Table 5 and 6. Emulators replicate hardware functionality, enabling firmware testing, protocol verification, and debugging on virtual devices. More challenging to require than simulators, they offer increased realism for the evaluation of dynamic and secure systems.

Table 5. Summarised LoRaWAN emulator and certification tools

Tool	Usability	Scalability	Flexibility	Licensing	Support	Device class	Integration
COOJA [37]-[39]	GUI & CLI for modelling; programmable via C (Contiki OS)	Small to large networks	Supports multiple devices, topologies	BSD License; citation required	Tutorials, mailing list, GitHub, user manual	A, B, C	NS-3, MATLAB, OMNeT++, FloRa, CupCarbon
ThingsBoard [9], [40]-[42]	Web-based interface for modelling & analysis	Highly scalable, multi-node support	Node programming in Java, C++, Python; supports OTAA, ADR, ABP	Apache 2.0 License	Installation guides, GitHub, tutorials	A, B, C	APIs for cloud/platform integration
LoRaWAN Emulator [43], [44]	GUI for building, simulating, and testing networks	Multi-device and multi-node setups	Supports network topologies and secure LoRaWAN protocols (OTAA, ADR, ABP)	Open-source; acknowledgement required	Help docs, videos, contact form	A, B, C	Integrates with external tools; secure communication support
LoRaWAN Gateway Emulator [28], [32]	Simulates endpoints & gateways; high message load	Multi-scenario capability	Configurable data rates, classes, frequencies	MIT License	GitHub repository for docs and issues	A, B, C	No external API or service integration
Packet Sender [45]	Gateway simulation with LoRaWAN packet generation	Handles high message volumes	Supports LoRaWAN 1.0.2 & 1.0.4; latency and power tuning	MIT License	GitHub repository	A, B, C	UDP packet forwarding; no API integration
MultiTech Conduit [46], [47]	Web interface for device/gateway management	Thousands of LoRaWAN devices	Supports LoRaWAN 1.0 & 1.1; live logging, frame/channel config	Commercial; free trial	Support portal, docs, forums	A, B, C	gRPC/REST APIs for cloud and DB integration
Semtech LoRa Cloud [48], [49]	Cloud service for device onboarding and monitoring	Global-scale deployments	adaptive data rates (ADR), geolocation, frequency hopping	Commercial subscription	Support portal and community forum	A, B, C	Cloud integration with databases and platforms
Actility ThingPark [15]	Network platform with full IoT connectivity suite	Public/private network support	LoRaWAN 1.0.2 & 1.0.4; custom API integration; supports LTE-M/NB-IoT	Commercial; free trial	Portal, community forum, marketplace	A, B, C	APIs for external services; cellular IoT support
MQTT Broker [26], [27]	MQTT-based server for client/device communication	Multi-tenant, scalable infrastructure	LoRaWAN-compliant; supports ADR and custom API/data handling	Open-source & commercial	Support portals, forums	A, B, C	TLS/SSL encryption, cloud, and DB integrations
LoRaTester [50]	CLI/GUI for LoRaWAN device certification	Supports diverse profiles	ADR & spreading factor support; physical device validation	Free to use	No dedicated portal; USB interface	A, B, C	No external API support; used for protocol compliance testing

Various devices and network topologies can be simulated by the BSD-licensed open-source COOJA platform. It is available both through command-line interfaces and graphical user interfaces. It is C programming-based with the Contiki OS and is greatly integrated with applications like MATLAB, NS-3, OMNeT++, FloRa, and CupCarbon. In addition, COOJA can realistically emulate security-critical scenarios and emulate all LoRaWAN device classes [37]–[39]. Other such features are the web-based UI that is scalable and Gatling for ThingsBoard performance testing under the Apache 2.0 license. It supports programming at the node level using Java, C++, and Python, and major LoRaWAN protocols like OTAA, ADR, and ABP. Role-based authentication, access controls, and encryption mechanisms are all part of its security model [9], [40]. The LoRaWAN Emulator offers a modular platform for network topology simulation with multiple nodes. Secure communication with all classes of devices is supported, and it operates based on the usage guidelines of the LoRa Alliance. In addition to duplicating network fault tolerance against possible attacks, this emulator may communicate with outside systems [1], [12], [24].

Similarly, Gateway Emulator supports Class A, B, and C devices, which are MIT-licensed open-source. Data rates, message types, and frequency settings are among the parameters being duplicated. It can send enormous volumes of data via UDP because it can be utilised both as an endpoint and as a gateway. Its functionality consists of authorisation, encryption, and authentication features for device management, but without any external services [1], [10], [41]. According to LoRaWAN 1.0.2 and 1.0.4 specifications, the Packet Sender, which is also an open-source implementation licensed under the MIT license, simulates endpoint and gateway functionality. It allows for stress testing by forwarding packets through UDP, which makes it possible to see how the network handles heavy traffic and whether throughput and reliability start to break down. It includes all types of devices and allows users to change performance settings, but it doesn't offer more advanced security features or easy API connections, [25]. The MultiTech Conduit provides a fully programmable LoRa gateway specifically designed for larger industrial IoT (IIoT) systems. It provides real-time frame logging, web interface, and faster dynamic channel switching. It is suitable for both LoRaWAN 1.0 and LoRaWAN 1.1 devices and offers REST APIs and gRPC, for easy connection to the cloud or a database. It also secures data with built-in authentication and encryption and tools.

Semtech LoRa Cloud are cloud-based management solution built on the LoRa Edge platform. The services range include certification testing, global coverage, and device location capabilities, global coverage with full support for Classes A, B, and C LoRaWAN devices. The Semtech LoRa CLoud adheres to the LoRaWAN 1.0.2 and 1.0.4 standards and can provide secure data management and device connectivity through integration with major cloud infrastructures. The security is enforced through standardized access control, authentication, and encryption to ensure that both device and data are reliably protected [48], [49]. In comparison, Actility's ThingPark platform provides LoRaWAN support through integrated IoT network management solution alongside cellular technologies such as LTE-M and NB-IoT. It covers the full device management lifecycle and handles network operations such as location services, data routing, and configuration management. It is fully compliant with LoRaWAN protocols by enabling easy integration with other systems through APIs. The architecture is designed for security and scalability, making it suitable for large-scale deployments such as IIoT, smart cities, and precision agriculture. While Semtech emphasizes edge capabilities and cloud-native services, and ThingPark focus more on network management and comprehensive lifecycle for complex, multi-technology environments.

The MQTT Broker component simulates server communication processes over the MQTT protocol and provides multi-device and multi-gateway communication. The module supports connectivity with cloud services as well as other APIs and has flexible architectures that can be set up with either open-source platforms or commercial setups. The broker has features such as secure data transmission utilizing TLS/SSL communication techniques and ACLs with authenticate methods that are based on credentials. The broker supports every class of device within the LoRaWAN protocol.

In this group, there is LoRaTester that is intended for use within the testing for conformance and inter-operability with devices that comply with the regional and international standards set for LoRaWAN technology. The device supports all device classes as well as ADR and spreading factors. Though technical expertise is required to set up the use of this technology, the platform provides further opportunity to improve on the security measures with customizable parameters and data protection policies [10].

The reviewed studies indicate that there exists a structured process development environment in terms of simulation and emulation with each one favoring separate aspects within the validation process of the IoT system simulation environments such as LoRaSim NS-3and flora enable an affordable and convenient validation process within the emulator environment to dynamically assess the operability within the LoRaWAN system these simulation environments are most favorable during initial stages with in-depth parameter adjustment availability during network setup and protocol analysis; however, these environments are incapable of simulating real-world environments with inherent innate constructs within the hardware environment during initial development stages the emulator environments such as the LoRaWAN emulator

and gateway emulator act as an aid to overcome these difficulties within these emulator environments with synchronized real-time node-to-gateway communication within these platforms the entire firmware validation process; the process of testing various parameters with possible renegotiation to ADRs; can be embodied within real-time operations with no overhead during these development stages the industrial platforms include TTN, Loriot, and Actility ThingsPark these platforms act as production platforms with standardized and encrypted parameters with built-in device management; gateway management; secure data transactions; and cloud analytics though these platforms yield scalable construct with reliable operations during these stages the platforms are incapable with construct diffusivity within the protocol during initial development stages these above-mentioned simulation environments; emulator platforms; and industrial platforms act as a collaborative ecological systematic environment simulation environments are conducive during theoretical validation processes emulator platforms are conducive during practical validation processes industrial platforms are conducive during operational stages these platforms act as an ideal holistic ecosystem with no redundancy during these stages.

Table 6. Summarised limitations of LoRaWAN emulator and certification tools

Tools	Performance	Metrics	Infrastructure	Device Support	Services	Security
COOJA [37]-[39]	Effective for Contiki-based simulation	Packet delivery, latency, energy usage, simulation time	Java-based GUI/CLI; requires Contiki OS	A, B, C	Network behavior testing, topology modelling	Basic encryption in simulation; customizable via Contiki stack
ThingsBoard [9], [40], [42]	High-performance telemetry and visualization	Uptime, flow rate, node latency, connectivity metrics	Scalable web platform; supports clustered deployments	A, B, C	Rule engine, node management, real-time data processing	TLS encryption, role-based access, token/cert-based authentication
LoRaWAN Emulator [43], [44]	Efficient LoRaWAN simulation, moderate complexity	Node interactions, packet status, message delivery	Web architecture: gateway-free	A, B, C	ADR/OTAA/ABP modelling, join testing	Supports secure channel simulation and protocol compliance testing
Gateway Emulator [28], [32]	Handles message-heavy scenarios	Gateway latency, throughput, class switching	Local setup: simulates gateways and devices	A, B, C	Gateway simulation, flow modelling	No integrated security; primarily for functional testing
Packet Sender [45]	Reliable packet simulation and crafting	Success rate, latency, IP/port delivery	Local tool using UDP; no cloud backend	1.0.2 & 1.0.4 devices	Manual packet testing	No encryption/authentication; external setups needed
MultiTech Conduit [46], [47]	Optimised for gateway traffic and protocol handling	Uptime, traffic, reception rate, latency	VPN/cloud-connected gateway infrastructure	A, B, C	Routing, cloud integration, gateway management	Secure gRPC/REST APIs, encryption, authentication
Semtech LoRa Cloud [48], [49]	Scalable, global coverage with strong telemetry	Battery life, MAC commands, traffic volume	Cloud platform supports LoRa Edge devices	A, B, C	Device tracking, MAC provisioning, modulation	Strong authentication, secure onboarding, and communication
Actility ThingPark [15]	High-performance platform for public/private networks	Latency, uptime, traffic analytics, dashboard metrics	Hosted/on-premises; supports LTE-M/NB-IoT	A, B, C	Advanced analytics, device management, integration marketplace	End-to-end encryption, identity management, authentication
MQTT Broker [26], [27]	High-throughput messaging and middleware integration	Broker latency, access logs, message loss rate	Middleware integrates via HTTP/UDP with servers/gateways	A, B, C	LoRaWAN stream handling, cloud storage support	TLS/SSL, access control lists, client authentication
LoRaTester [50]	Precision testing for device certification	Power use, signal range, transmission efficiency	USB-based, standalone test tool	Protocol-compliant devices	Firmware update simulations, activation tests	Security validation supported; lacks full encryption/enforcement

2.3. Summary of identified research gaps

Although different LoRaWAN tools are available, only a few comprehensive and focused security comparisons exist. Existing studies often take a narrow view, focusing on performance evaluation-such as latency or throughput-without systematically investigating integration, scalability, and security features of simulators, emulators, and infrastructure platforms. Most of the tools also lack unified frameworks for evaluation that tie quantitative scoring together with expert validation. What is needed is an approach that integrates technical assessment with visual synthesis, such as heatmaps or radar charts, which will enable both researchers and practitioners to identify the most suitable tool for particular stages of IoT system design and deployment.

The existing literature on LoRaWAN development tools still indicates the gap between simulation-based experimentation and deployment-oriented evaluation. Whereas previous works focused on particular topics, such as network scalability, protocol optimization, or usability of a tool, a comprehensive security-aware comparative analysis remains limited. This review extends the current knowledge by systematically classifying and evaluating LoRaWAN simulation, emulation, and deployment tools within a unified analytical framework. It hence provides, for the first time, a structured basis for researchers and practitioners in finding appropriate platforms for both experimental and real-world IoT implementations. While individual LoRaWAN simulators or deployment platforms have been addressed in past studies, a unified security-centric comparative framework remains limited. Building upon the research gaps identified in the literature the following section presents the methodology adopted to systematically assess 26 LoRaWAN related tools across three primary dimensions scalability integration and security.

3. METHOD

This study employs a structured approach to methodologically assess 26 simulation-emulation platforms for scalability integration capacity and security performance throughout the research process the analysis was segmented into five stages that include data collection categorization of the data collection tool validation analysis with subsequent visualization to establish methodological rigor the process is explained below with a conceptual perspective outlined in Figure 2.

- Data collection: The data collection was initiated with an analysis involving state-of-art searches on various academic databases such as IEEE Xplore, ScienceDirect, ACM DL, with Google Scholar to identify the most pertinent simulation and emulation tools with respect to LoRaWAN. While undertaking the research process to optimize the search results with respect to LoRaWAN, various criteria had to be met to ensure that the most pertinent simulation and emulation tools are taken into account. A set criteria was therefore set to ensure validity with respect to essential communication parameters such as OTA, ABP, and ADR, with simulation/emulation capabilities that are device class A, B, and C compliant with either fundamental or advanced security. A list of 16 simulation and 10 emulation platforms was adopted.
- Tool classification: The analyzed data was then classified to identify the various categories that the final set of tools belong to. This was carried out with the use of different dimensions. The dimensions are set to ensure that the best possible data is derived through peer review processes and forums. A structured list was followed to identify these dimensions that include analysis from peer review sites and other official platforms. The dimensions that are used include (1) usability reviewing the usability features such as setup simplicity and interface simplicity; (2) scaling reviewing flexibility; (3) configuration; (4) protocol modifiability reviewing whether the licensing is free/open-source or copyrighted; (5) reviewing support and tutorial simplicity; (6) reviewing compatibility with classes A, B, and C; (7) reviewing whether MATLAB simulation is possible within the platforms such as OMNET++, NS-3, and MATLAB; these dimensions ensure that there are no ambiguities in understanding the usability features of the various tools that are identified.
- Comparative evaluation: The purpose of the Comparative evaluation stage was to compare the usability scalability and security capabilities of each candidate tool. A hybrid evaluation method was adopted that entailed carrying out a qualitative analysis in conjunction with semi-quantitative scoring that concentrated on utilizing a five-point scoring scale ranging from 1 to 5. These factors differed depending on whether the candidate technology was a simulation platform or an emulator; these platforms were rated on a scale of 0 to 5 with respect to the application of encryptions and authentication processes and protocol-level security; simulation platforms rated on a three-point scoring scale that denoted the level of securities with 1 symbolizing low levels, 2 indicating medium levels, and 3 symbolizing strong levels with respect to the application of encryptions and adherence to LoRa WSN securities.

- The focus is on two perspectives that framed the research relevance with regard to adaptability with respect to testing experimentation and deployment-readiness with regard to scalability and interoperability. The results that are integrated within integrated security are presented in tables
- Validation: a process of validation by triangulation has been adopted; the results obtained are tested with normative research studies referred to similar comparative analysis in the past and reports from the concerned developer organizations; this step has therefore improved internal validity and minimized the impact of the evaluator's bias; peer review articles are further referred to ensure that there are no mismatches between the observed attributes and the claimed results; this step has made the study free from invalid methodologies and added to the reliability of the conclusions made to propose the evaluation framework. This validation step has thus added to the validity of the proposed evaluation frame.
- Visual synthesis: the visual synthesis process was adopted on the justified results to ensure that there was an easy means of understanding and making decisions from the data. The data was presented in visual form through various means such as performance tables, heatmaps, and radar charts. It aids to understand the strengths and weaknesses of the different emulators and emulators presented in this paper.

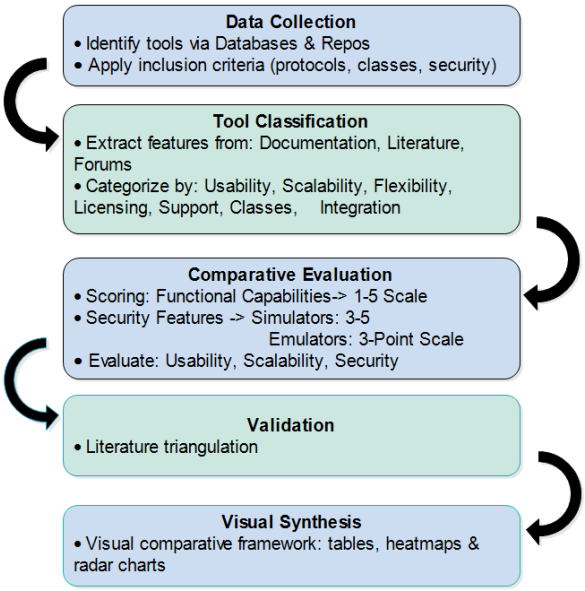

Figure 2. Tool's selection and evaluation process

4. RESULTS AND DISCUSSION

In this section, we explore how different LoRaWAN frameworks simulation, emulation, and full deployment stack up, combining both qualitative insight and quantitative analysis. Simulation tools, such as NS-3 or LoRaSim, are often the foremost preference for researchers. They are inexpensive, easy to change, and quite flexible for testing new ideas or checking early performance. For example, you can try out a new way of adjusting data rates or test thousands of fake sensors without using any real ones. Simulation makes all that possible. However, there's a downside: these tools don't fully show how real networks work. They can't capture things like interference from other nearby devices or random packet loss that happen in real life. Emulation tools and deployment setups go beyond simulation. Platforms like FLoRa or cloud-based testbeds let you test networks on a bigger scale, often close to real conditions. They offer better security, handle different types of devices, and work well with other systems. This makes them useful for moving from ideas to real use. Developers can check if a protocol that works in simulation still works when things get complicated in the real world. In short, simulation is great for controlled tests and learning, but emulation and deployment tools are necessary if you want to be sure your LoRaWAN solution can handle the real world.

4.1. Results

4.1.1. LoRaWAN simulation tools

Simulation tools are necessary to model, test, and validate network behavior before actual deployment. They facilitate a cost-effective and flexible environment in which to assess performance

indicators related to latency, energy consumption, packet delivery, and scalability under different configurations. For instance, COOJA, NS-3, and CupCarbon allow researchers and developers to investigate LoRaWAN protocol operations, such as the optimization of network parameters and performance evaluation at a large scale, which is of the highest importance. They are very helpful for the study of environmental factors, mobility, and interference effects on communication efficiency.

Figure 3 shows the comparison of 16 LoRaWAN simulation and deployment tools in terms of five core features: usability, scalability, flexibility, support, and integration. TTN, Loriot, and NS-3 perform very well across a range of scalability and integration issues, indicating their wider applicability. LoRaWAN Test Suites and LoRaSimu rank lower in support and extensibility.

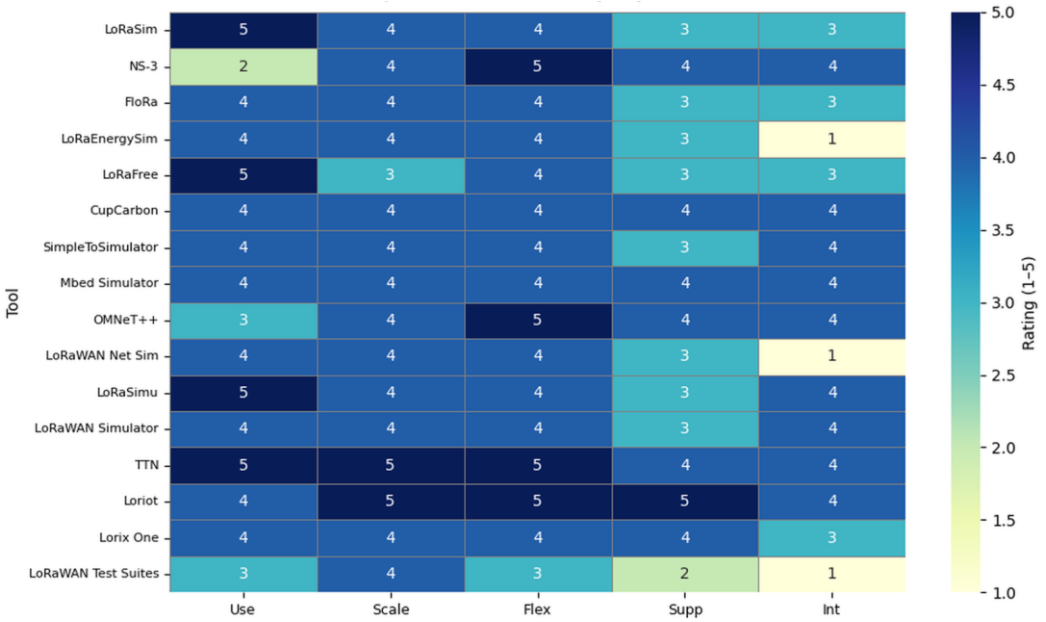

Figure 3. Comparison of LoRaWAN tools features

Figure 4 depicts the security capabilities of these tools. Deployment platforms like TTN and Loriot achieve the highest scores due to integrated encryption, authentication, and access control; simulators like NS-3 and OMNeT++ score lower, with an emphasis on modeling over enforcing security measures. This means that while the simulators are very suitable for performing research or trying out something, deployment platforms perform better in a secure, production-level environment.

However, even the most useful simulators have their weaknesses. They are highly dependent on model assumptions, breakdown under high traffic, usually ignore hardware constraints, and may support only partial protocol features. Again, they require specialized technical knowledge, and some involve licensing costs.

4.1.2. LoRaWAN emulator tools

Emulators offer closer-to-real conditions for testing hardware and network interactions. Whereas simulations work with some sort of abstraction, emulation reproduces real transmission behavior, device interaction, and gateway communications in near real time. They enable developers to perform firmware testing, protocol compliance validation, and system performance evaluation under controlled conditions. Other tools worth mentioning include LoRaWAN Emulator, Gateway Emulator, and MultiTech Conduit that are useful in performing deployment readiness, device interoperability, and certification testing.

Figure 5 presents a comparison of the features of ten different emulation and cloud-based tools. ThingsBoard, MultiTech Conduit, Semtech LoRa cloud, and actility ThingPark consistently present high scores with regards to scalability, flexibility, and integration, which all make them appropriate for large-scale deployments. By contrast, LoRaWAN gateway emulator, Packet Sender, and LoRa tester present limited integration and support, indicating that they are much better suited for specific tasks rather than full system simulations or deployments.

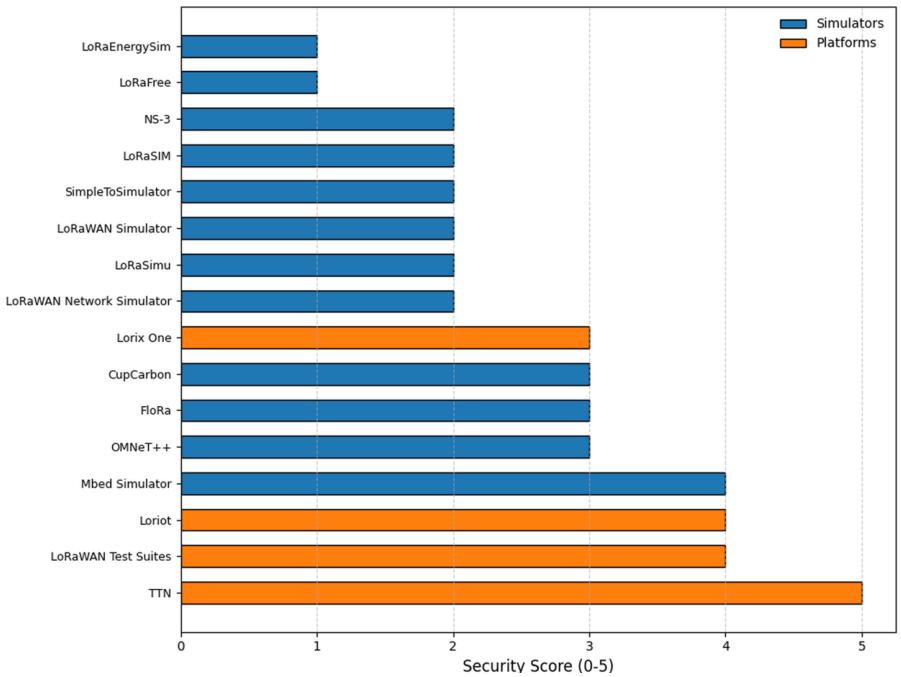

Figure 4. Security capabilities across all tools, real-world platforms

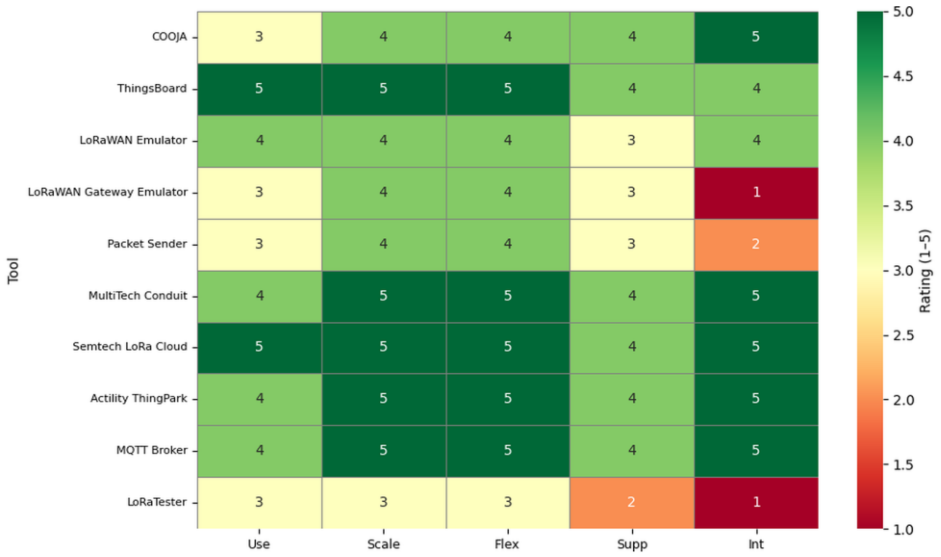

Figure 5. Comparison of LoRaWAN emulation and cloud tools by important features

Figure 6 compares their security capabilities. We used a 3-point scale (1 = Weak, 2 = Moderate, 3 = Strong), focusing on encryption, authentication, and compliance. As shown, tools such as ThingsBoard, ThingPark, and Semtech LoRa cloud offer robust encryption (e.g., TLS), strong authentication (e.g., certificates, RBAC), and full standards compliance. This makes them suitable for secure, production-level deployments. In addition, moderate tools like MQTT Broker and COOJA show strengths in specific areas (e.g., encryption or simulation) but fall short in compliance or authentication. Thus, limiting them to testing or non-critical use. Tools like Packet Sender and LoRa tester score poorly across all axes due to missing basic security features and should be used only in controlled environments. Figure 6 shows overall security posture; larger, balanced shapes indicate stronger tools, while asymmetries reveal trade-offs. It aids decisions but simplifies real-world complexity; added dimensions like audit logging could enhance it.

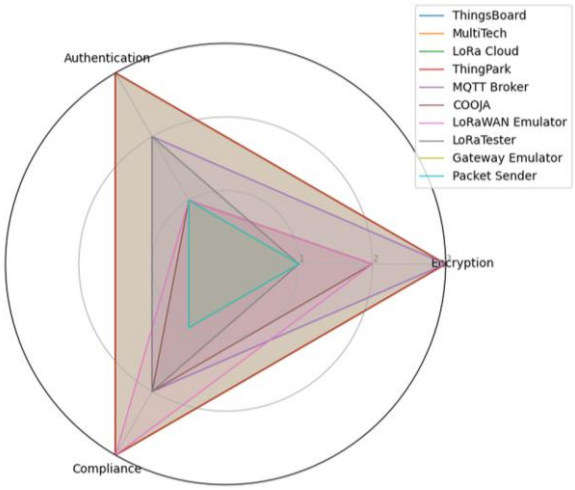

Figure 6. LoRaWAN emulators security comparison

4.2. Discussion

From Tables 1-6, it can be observed that scalability, protocol support, device compatibility, and security requirements have been major points of interest in the selection of a tool for LoRaWAN developmentSimulation environments like LoRaSIM [1], [2], [5], [12], NS-3 [17], and FloRa [16], [21]-[23], provide cost-effective, high-performance modelling for packet delivery, latency, and energy consumption. Accordingly, NS-3 stands out for its protocol flexibility and integration with C++ and Python-based scripting [5]-[7], and coupling with OMNeT++ and SimuLTE offers broader urban IoT simulation capabilities [21]-[23]. Despite their strengths, while FloRa's simulation tools often lack native encryption modules and require manual implementation of TLS or DTLS protocols. As shown in Table 5, emulators like LoRaWAN Emulator [43], [44]. Gateway emulator [28], [32] and Packet Sender [45] offer real-time testing under more realistic conditions, simulating node interactions, message throughput, and ADR/OTAA behavior. Tools like LoRaWAN Emulator support secure channel modelling and protocol compliance, but others like Packet Sender emphasize functional packet generation without integrated encryption. However, some, like Packet Sender [45], lack integrated encryption. Infrastructure platforms like TTN [6], [12], [30], Loriot [10], [20], [31], and ThingPark [15], deliver enterprise-grade performance with extensive support for gateway connectivity, cloud integration, and device provisioning. TTN supports open-source scalability and secure join mechanisms via AES encryption and access keys. In this case, emulators like ChirpStack [9], [35], [36] and ThingsBoard [9], [40], [42] bridge this gap with support for TLS/SSL, OAuth2, and certificate-based authentication. Scalability and integration also show clear distinctions. Platforms like MultiTech Conduit [46], [47] and Semtech LoRa Cloud [48], [49] handle thousands of end nodes, support REST/gRPC APIs, and offer geolocation services. By contrast, simulators operate in isolated environments with limited real-world device interaction. In practice, choosing the right LoRaWAN tool depends on the specific stage of development, the level of modelling detail required, and the project's security and scalability needs. Thus, a combined approach, using simulators for protocol design, emulators for system behaviour testing, and infrastructure platforms for deployment, provides a practical and flexible approach for building dependable, secure LoRaWAN-based IoT systems.

There are, of course, several trade-offs to consider when considering simulators and emulators for LoRaWAN applications. Simulators are more ideal and flexible for LoRaWAN experiments; however, they often lack full compliance to replicate LoRaWAN real-world interactions. On the other hand, emulators can often be deployed to bridge the gap by creating a more realistic LoRaWAN testbeds, however, in some instances, they can be more resource intensive. Similarly, infrastructure platforms, can be deployed to prioritize scalability and security, but they can be costly concerning iterative design or rapid prototyping. Understanding these distinctions between emulators and simulators can aid researchers and developers to make informed choices, selecting tools that match the specific phase, scale, and security demands of a given LoRaWAN-based IoT project.

4.3. Future research directions and limitations

This subsection highlights some of the identified future research directions. Despite the valuable capabilities of current LoRaWAN simulation and emulator tools, several limitations call for further investigation.

- Scalability and accuracy remain significant challenges; existing simulators often struggle to replicate real-world conditions, particularly in large-scale IoT deployments affected by dynamic interference, channel contention, and variable energy profiles. Future models should incorporate adaptive channel allocation, realistic interference simulations, and power optimization frameworks to improve fidelity and responsiveness.
- Expanding LoRaWAN's reach into non-terrestrial networks, such as hybrid satellite-UAV-terrestrial systems, is another promising frontier. Interfacing LoRaWAN tools with concepts from recent advancements in integrated space-based communication, including ergodic capacity modelling for NOMA overlays, could unlock new avenues for remote and space-driven IoT applications. Emulator platforms designed for hybrid terrestrial-satellite scenarios will be crucial for prototyping such deployments.
- Security, though increasingly prioritized, still lacks uniform, advanced support. Most tools are limited in their implementation of modern cryptographic protocols, intrusion detection systems, and resilience against active threats like jamming. Integrating post-quantum cryptography, AI-driven anomaly detection, and adaptive security modules could significantly strengthen defenses across simulation and emulation environments.
- On the energy front, while tools like LoRaEnergySim enable consumption profiling, there's still a gap in real-time adaptive energy harvesting and optimization techniques. Embedding machine learning models for dynamic energy efficiency could prolong sensor node lifespans and enhance network sustainability, especially in remote and resource-constrained deployments.
- Interoperability remains a critical area. Many LoRaWAN tools still operate in isolation, limiting cross-platform evaluations and collaborative development. Creating standardized APIs and middleware frameworks would facilitate data exchange and workflow integration, allowing researchers and practitioners to combine simulation depth with emulator realism for more comprehensive testing and deployment.

5. CONCLUSION

This paper presented a comprehensive comparative analysis of current LoRaWAN simulation and emulator tools, evaluating their usability, scalability, security features, licensing, and integration capabilities. Our findings show that simulation tools such as NS-3, OMNeT++, and LoRaSim offer flexible, cost-effective environments suited for academic and protocol-level research, enabling evaluations of latency, packet delivery, and energy consumption across device classes A, B, and C. However, their accuracy is constrained by simplified environmental modeling and often lacks built-in support for end-to-end security. Conversely, emulator platforms such as TTN, ChirpStack, Loriot, and Lorix One provide high-fidelity testing under real-world conditions, supporting scalable deployment, secure communications, and integration with gateways and cloud infrastructure. Tools like MultiTech Conduit and ThingPark extend these capabilities through advanced telemetry, robust device provisioning, and lifecycle management. Infrastructure platforms and visualization tools, including COOJA, ThingsBoard, and LoRaWAN Emulator, bridge simulation and implementation by offering real-time telemetry, GUI-based modelling, and varying levels of security. While many of these platforms now support protocols such as TLS and OAuth2, simulation environments still often depend on manual security configurations. These findings underscore the need for more adaptable and secure LoRaWAN tools that incorporate AI-driven intrusion detection, cryptographic enhancements, and interoperability frameworks to unify simulation and emulation efforts. The distinctions between simulation, emulation, and infrastructure platforms reflect the different stages of network design, testing, and deployment. A hybrid strategy that combines simulators for design, emulators for validation, and infrastructure platforms for real-world implementation is recommended for developing robust, scalable, and secure LoRaWAN networks. Future research should focus on improving simulation accuracy, satellite integration, and adaptive energy modeling. Also, standardized evaluation methods, especially HIL testing, are essential to ensure LoRaWAN tools remain reliable and relevant for emerging IoT applications.

FUNDING INFORMATION

This was supported by FNAS, UDSC, and the Department of Computer Science at the North-West University, Mafikeng campus, as well as the Council for Scientific and Industrial Research (CSIR) via the Smart Networks collaboration initiative and IoT-Factory Program (funded by the Department of Science and Innovation (DSI), South Africa).

AUTHOR CONTRIBUTIONS STATEMENT

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Ntshabele Koketso	✓	✓		✓		✓	✓	✓	✓					
Isong Bassey	✓		✓	✓			✓	✓		✓	✓	✓	✓	✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] L. Alliance, "Online simulation of LoRaWAN™ devices." https://www.youtube.com/watch?v=NJyGYh2WeZ8 (accessed Jun. 06, 2025).
[2] G. Bernardinetti, F. Mancini, and G. Bianchi, "Disconnection attacks against LoRaWAN 1.0.X ABP devices," 2020, doi: 10.1109/MedComNet49392.2020.9191495.
[3] M. Hamnache, R. Kacimi, and A. L. Beylot, "Unifying LoRaWAN networks by enabling the roaming capability," in *Proceedings - Conference on Local Computer Networks, LCN*, 2021, vol. 2021-October, pp. 371–374, doi: 10.1109/LCN52139.2021.9524996.
[4] K. Q. Abdelfadeel, D. Zorbas, V. Cionca, and D. Pesch, "FREE - Fine-grained scheduling for reliable and energy-efficient data collection in LoRaWAN," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 669–683, 2020, doi: 10.1109/JIOT.2019.2949918.
[5] R. Marini, K. Mikhaylov, G. Pasolini, and C. Buratti, "Lorawansim: A flexible simulator for lorawan networks," *Sensors (Switzerland)*, vol. 21, no. 3, pp. 1–19, 2021, doi: 10.3390/s21030695.
[6] N. Blenn and F. Kuipers, "LoRaWAN in the Wild: Measurements from the things network," 2017, [Online]. Available: http://arxiv.org/abs/1706.03086.
[7] B. Thébaudeau, "contiki-os/contiki," 2019. https://github.com/contiki-os/contiki/wiki/ (accessed Jun. 06, 2023).
[8] A. Grunwald, M. Schaarschmidt, and C. Westerkamp, "LoRaWAN in a rural context: Use cases and opportunities for agricultural businesses," in *24. ITG-Symposium on Mobile Communication - Technologies and Applications*, 2020, pp. 134–139.
[9] ThingsBoard, "ThingsBoard getting started." https://www.chirpstack.io/docs/guides/thingsboard.html (accessed Jun. 06, 2023).
[10] M. Afhamisis, S. Barillaro, and M. R. Palattella, "A testbed for LoRaWAN Satellite Backhaul: Design principles and validation," *2022 IEEE International Conference on Communications Workshops, ICC Workshops 2022*, pp. 1171–1176, 2022, doi: 10.1109/ICCWorkshops53468.2022.9814560.
[11] A. Valente, S. Silva, D. Duarte, F. C. Pinto, and S. Soares, "Low-cost lorawan node for agro-intelligence iot," *Electronics (Switzerland)*, vol. 9, no. 6, 2020, doi: 10.3390/electronics9060987.
[12] T. T. Network, "LoRaWAN." https://www.thethingsnetwork.org/docs/lorawan/ (accessed Jun. 12, 2023).
[13] A. Bounceur *et al.*, "CupCarbon: A new platform for the design, simulation and 2D/3D visualization of radio propagation and interferences in IoT networks," *CCNC 2018 - 2018 15th IEEE Annual Consumer Communications and Networking Conference*, vol. 2018-January, pp. 1–4, 2018, doi: 10.1109/CCNC.2018.8319179.
[14] M. Piechowiak and P. Zwierzykowski, "Simulations of the MAC layer in the LoRaWAN networks," *Journal of Telecommunications and Information Technology*, vol. 2, pp. 22–27, 2020, doi: 10.26636/JTIT.2020.144120.
[15] ThingPark®, "ThingPark® - heartbeat of your IoT ecosystem." https://www.thingpark.com/ (accessed Aug. 14, 2024).
[16] J. D. Yi Loor, A. Espinal, and V. Sanchez Padilla, "Lorawan-based RSSI-trilateration model for node location: A simulation integrating FloRa and Omnet++," *Transport and Telecommunication*, vol. 25, no. 2, pp. 218–229, 2024, doi: 10.2478/ttj-2024-0017.
[17] Ns-3, "ns-3 network simulator." https://www.nsnam.org/ (accessed Jun. 06, 2023).
[18] S. Idris, T. Karunathilake, and A. Förster, "Survey and comparative study of LoRa-enabled simulators for internet of things and wireless sensor networks," *Sensors*, vol. 22, no. 15, 2022, doi: 10.3390/s22155546.
[19] M. A. M. Almuhaya, W. A. Jabbar, N. Sulaiman, and A. H. A. Sulaiman, "An overview on LoRaWAN technology simulation tools," *Lecture Notes on Data Engineering and Communications Technologies*, vol. 127, pp. 345–358, 2022, doi: 10.1007/978-3-030-98741-1_29.
[20] F. Hofer and C. Kuen, "Off-the-shelf LoRaWAN: Experimenting on the prospect of a low-cost rapid prototyping solution," *Proceedings - 2022 IEEE 46th Annual Computers, Software, and Applications Conference, COMPSAC 2022*, pp. 1020–1025, 2022, doi: 10.1109/COMPSAC54236.2022.00159.
[21] FloRa, "A framework for LoRa simulations with OMNeT++," 2022. https://flora.aalto.fi/ (accessed Jun. 06, 2023).
[22] OMNeT++, "FLoRa (Framework for LoRa)." https://flora.aalto.fi/ (accessed on 1 July 2021). (accessed Jun. 06, 2023).
[23] OMNeT++, "Simulation models and tools," *OMNeT*, 2022. https://omnetpp.org/download/models-and-tools (accessed Jun. 06, 2023).
[24] UniCT-ARSLab, "LWN-simulator." https://github.com/UniCT-ARSLab/LWN-Simulator (accessed Jun. 06, 2023).

[25] E. B. Sanchez and D. F. H. Sadok, "Lora and lorawan protocol analysis using cupcarbon," *Communications in Computer and Information Science*, vol. 1280, pp. 352–376, 2020, doi: 10.1007/978-3-030-62554-2_26.
[26] K. Manditereza, "LoRaWAN and MQTT." https://www.hivemq.com/article/lorawan-and-mqtt-integrations-for-iot-applications-design/ (accessed Jun. 27, 2023).
[27] E. Ouanounou, "LoRaWAN and MQTT." https://www.cyberark.com/resources/threat-research-blog/lorawan-mqtt-what-to-know-when-securing-your-iot-network (accessed Jun. 27, 2023).
[28] SimpleSoft, "IoT simulation for LoRaWAN." https://www.simplesoft.com/SimpleIoTSimulatorForLoraWan.html (accessed Jun. 06, 2023).
[29] ArmMBED, "Introducing the Mbed simulator." https://os.mbed.com/blog/entry/introducing-mbed-simulator (accessed Jun. 06, 2023).
[30] T. T. Stack, "The things stack documentation." https://www.thethingsindustries.com/docs/the-things-stack/cloud/ (accessed Sep. 05, 2023).
[31] LORIOT, "Enabling massive IoT worldwide." https://www.loriot.io/ (accessed Sep. 05, 2023).
[32] T. T. Industries, "The things stack." https://www.thethingsindustries.com/docs/gateways/models/wifx-lorix-one/lbs/ (accessed Jul. 03, 2023).
[33] W. Sàrl, "LORIX one." https://www.lorixone.io/ (accessed Jun. 26, 2023).
[34] KeySight, "LoRaWAN test challenges." https://www.keysight.com/us/en/assets/7018-05937/application-notes/5992-2595.pdf (accessed Jun. 06, 2023).
[35] ChirpStack, "ChirpStack, open-source LoRaWAN® network server." https://www.chirpstack.io/ (accessed Aug. 21, 2023).
[36] Openremote, "Tutorial: Receive LoRaWAN sensor data from ChirpStack," 2025. https://github.com/openremote/openremote/wiki/Tutorial%3A-Receive-LoRaWAN-sensor-data-from-ChirpStack.
[37] Contiki-ng, "Cooja." https://github.com/contiki-ng/cooja.
[38] Contiki-NG Maintainers, "Running Contiki-NG in Cooja — Contiki-NG documentation." https://docs.contiki-ng.org/en/develop/doc/tutorials/Running-Contiki-NG-in-Cooja.html.
[39] N. S. Tools, "Cooja simulator for IoT download." https://networksimulationtools.com/cooja-simulator-for-iot-download/ (accessed Sep. 26, 2023).
[40] ThingsBoard, "ThingsBoard open-source IoT platform." https://thingsboard.io (accessed Jun. 06, 2023).
[41] Kartben, "lorawan node simulator." https://github.com/kartben/lorawan-node-simulator (accessed May 31, 2023).
[42] ThingsBoard, "ThingsBoard." https://thingsboard.io/.
[43] I. Akanova, D. Urazayev, Y. Kadirzhanov, and D. Zorbas, "Demo: A LoRaWAN emulator testbed," *Proceedings - IEEE Symposium on Computers and Communications*, vol. 2023-July, 2023, doi: 10.1109/ISCC58397.2023.10218138.
[44] M. Gaffurini, A. Flammini, P. Ferrari, D. Fernandes Carvalho, E. P. Godoy, and E. Sisinni, "End-to-end emulation of LoRaWAN architecture and infrastructure in complex smart city scenarios exploiting containers," *Sensors*, vol. 24, no. 7, 2024, doi: 10.3390/s24072024.
[45] D. Nagle, "Packet sender," 2014. https://packetsender.com/ (accessed Aug. 07, 2024).
[46] M. S. Inc, "Conduit® hardware guide." https://multitech.com/wp-content/uploads/S000690-Conduit-Hardware-Guide.pdf (accessed May 17, 2023).
[47] MULTITECH, "Access point for LoRa® technology Conduit® AP MTCAP & MTCAP2 series." https://multitech.com/all-products/cellular/cellular-gateways/conduit-ap/ (accessed Feb. 05, 2024).
[48] Semtech, "LoRa coud services have relocated." https://info.semtech.com/lora-cloud-geolocation-service (accessed Jun. 06, 2023).
[49] Semtech, "Platform for IoT," *Semtech Corporation*, 2015. https://www.semtech.com/lora/what-is-lora (accessed Jun. 06, 2023).
[50] HARDWARIO, "New lora tester." https://github.com/hardwario/twr-lora-tester (accessed Aug. 07, 2024).

BIOGRAPHIES OF AUTHORS

Dr. Ntshabele Koketso earned a Bachelor of Science degree in computer science and mathematics from North-West University's Mafikeng Campus in Mafikeng, North-West; in 2017; he earned an Honours Bachelor of Science degree in computer science from the same institution; and in 2019, he earned an MSc in computer science from the same institution. He received his PhD in computer science from North-West University's Mafikeng Campus in Mafikeng, North-West, in 2025. He served as a student researcher from 2016 to 2023, publishing several papers under the FRC and partially through the Department of Computer Science at NWU-Mafikeng and CSIR, South Africa. Cognitive Radios, data security in LPWAN, CR-LPWAN, and energy efficiency in LoRaWAN and IoT are just a few of his research interests. In 2018, he worked at the university as a laboratory technician, a position he maintained until March 2019. He is currently as doctor and a permanent lecturer in the Department of Computer Science. He can be contacted at email: koketso.ntshabele@nwu.ac.za.

Prof. Bassey Isong received his BSc. Degree in Computer Science from the University of Calabar, Nigeria in 2004, MSc degrees in Computer Science and Software Engineering from the Blekinge Institute of Technology, Sweden in 2008 and 2010 respectively, and a PhD degree in Computer Science from the North-West University in 2014. His research interests include and are not limited to Software Engineering, Software Defined Networks, Smart Networks, Machine Learning, and Blockchain. He is currently a Professor in the Department of Computer Science at North-West University, Mafikeng Campus, South Africa. He is also a member of the IEEE Computer, Communication, and Education Societies as well as ACM. He can be contacted at email: bassey.isong@nwu.ac.za.