International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(20s) 2026 pp. 1811-1836
https://doi.org/10.70917/ijcisim-2026-1887
Article

Blockchain-Enabled Fault-Tolerant IoT Intrusion Detection Using Spatio-Temporal Deep Learning

Jyotsna Tarigoppula[1], Dr. K. Madhuri[2]

[1] Assistant Professor, Anurag University, Hyderabad., Email: jyotsna.akkinapalli@gmail.com
[2] Associate Professor, Anurag University, Hyderabad.

Abstract: Due to the swift expansion of the IoT networks, there are some critical security issues such as node compromising, data manipulation, and Sybil attacks. This paper presents a fault-tolerant IoT security architecture based on blockchain and multi-layer anomaly detection with secure consensus-based systems. The IoT devices are authenticated via a blockchain-based identity verification and through cryptographic authentication to verify legitimate participation, sensor data is encrypted, timestamped and logged into blockchain to prevent tampering. Data cleaning, normalization, and outlier elimination are also used as preprocesses to the ToN-IoT dataset in order to improve the quality of the data. Temporal and statistical characteristics are derived to identify time series trends as well as cross-node associations and a new Spatio-Temporal Reinforced Swarm Optimization (ST-RSO) algorithm is used to select the most pertinent features, lowers the dimension, and enhances detection rates. The detection of anomalies is carried out with the help of a Multi-Scale Residual Graph Attention Network (MS-RGAN), which captures both local and global interactions among devices with the flow of time. Verification Alerts are validated with a Majority Voting Ensemble (MVE) to minimize the number of false positives and verified events are stored on the blockchain with PBFT consensus, which ensures decentralized and resistant to tampering decision-making. Nodes that are compromised are automatically isolated, updated on trust scores and reintegrated after verification. The proposed MS-RGAN model has a high level of performance with a high accuracy of 98.7, a high precision of 98.4, high recall of 98.6, and F1-score of 98.5 and a very low FPR of 0.9 and FNR of 1.2. It has a high MCC of 96, 98.2% NPV, lower detection latency (21 ms), 41.8% feature drop, and very low consensus validation time (14.2 ms) hence it is highly efficient and reliable in intrusion detection on the IoT. The review shows that the framework can be used to achieve high detection, low false alarms, and robust performance in the presence of distributed IoT threats.
Keywords: Blockchain-Enabled Security, IoT Intrusion Detection, Spatio-Temporal Deep Learning, MS-RGAN, Fault Tolerance, Feature Reduction, Low-Latency Detection, Consensus Validation

1. Introduction

Internet of Things (IoT) has transformed the lives of people in the modern world by guaranteeing convenient interconnectedness amid devices, sensors, networks in various sectors such as healthcare, smart cities and industrial automation. However, there were some severe security threats that were posed by the high IoT devices development rate [1]. The majority of the IoT devices have modest computing capabilities, weak authentication, and poor encryption, placing them within the perspectives of the cyberattacks, including, but not confined to, Distributed Denial of Service (DDoS), malware injection, and unauthorized access [2]. All these vulnerabilities are highly threatening both to individual devices and integrity and availability of entire IoT networks. One of the most significant defence mechanisms that have been applied in detecting malicious activities in the IoT networks is Intrusion Detection Systems (IDS) [3]. The traditional approaches of the IDS, such as signature-based and anomaly-based approaches, are centralized in processing and pre-established patterns. Despite the practical use of these approaches in some environments, they face challenges of dynamism, distribution and heterogeneity of IoT environments [4]. The

intensive use of devices, the diversity of devices, and real-time streams of data demand an IDS that has to be dynamic and capable of expanding and enduring failures [5]. One of the possible solutions to enhance the security of the Internet of Things is the use of blockchain technology to ensure its decentralization, impossibility of alteration, and transparency.

Blockchain will enable the safe exchange of information between the devices of the IoT recording the work of the equipment and its actions in unchangeable registers unreachable by a third party. Combining blockchain and IDS may result in increased trust, elimination of manipulation of the data, and accountability in distributed networks [6]. Also, using consensus mechanisms, blockchain can be used to organize detection notifications and responses among devices to support coordinated security without a single point of failure. Fault-tolerant intrusion detection in combination with blockchain makes the IoT systems resistant to attack [7]. Fault tolerance further implies that the IDS is resilient enough to remain functional when certain nodes go dead or become malicious, which is important when IoTs are deployed in large scale and the system is mission critical. The blockchain layer ensures data integrity and secure coordination and the IDS layer tracks network operations because of anomalies. This integration is expected to deliver greater detection, resistance to attacks, and a self-reinforcing security model to meet dynamic threat demands in changing IoT settings.

In spite of the fast growth of IoT implementation in critical infrastructures, current security measures are inadequate to manage large scale, decentralized, and dynamic digital attacks. Conventional intrusion detection systems do not have a capability of detecting cross-device attacks, temporal patterns, and orchestrating attacks like the Sybil, spoofing, and manipulating data [8]. Moreover, the majority of IoT systems are implemented on centralized structures, and they are prone to single-point failures, spoilage, and unauthorized access of nodes [9]. The existing feature selection and anomaly detection methods are also not effective in dealing with high-dimensional, noisy, and skewed data like ToN-IoT, which also trigger a high number of false positives and low levels of detection [10]. Thus, it is urgently required that it should have a decentralized, intelligent and fault-tolerant IoT security architecture that is able to authenticate devices, guarantee data integrity, refine feature relevance and reliably identify advanced attack patterns in real time [11]. This study is driven by the fact that there are rising dynamism of IoT-based automation in smart cities, healthcare, logistics, and industry systems where a single breached device can lead to massive operational collapse. Traditional machine learning [11] and rule-based systems [12] do not have the ability to generalize against threats that are unknown and provide minimal resiliency in dynamic IoT systems. Bringing blockchain can give a chance to decentralize trust, provide devices identity, which is tamper-proof and increase auditability, yet it must be combined with sophisticated AI-based anomaly detection to provide end-to-end security. The introduction of multifaceted data such as ToN-IoT makes it clear that intelligent selection of features and deep learning structures are needed to learn about spatio-temporal relationships and network-level behaviors. In that way, the given research will close the gap between trust enforcement provided by a blockchain and anomaly detection provided by a deep learning and offer a scalable and efficient anti-hacking protection model to the next-generation IoT environments.

1.1 Key contribution

Following are the key contribution of the study,

- All IoT nodes undergo secure enrolment using blockchain-based identity verification and cryptographic authentication, ensuring only trusted devices participate and preventing spoofing, Sybil, or unauthorized access.
- The proposed framework includes multiple security stages: secure authentication, feature extraction, feature optimization, anomaly detection, alert validation, and trust-aware fault recovery to achieve full-cycle IoT protection.
- In order to achieve this, the ToN-IoT dataset is evaluated, and all sensor data are systematically pre-processed, which includes cleaning, normalization and outliers' removal so that the data is consistent and the noise effect is reduced.

- To identify insights to the pattern of device behavior, both temporal and statistical feature extraction methods are used, including the measure of variance, skewness, network activity trends, and cross-node dependencies to represent the enriched anomalies.
- One of the new features is the introduction of a novel feature optimization algorithm named Spatio-Temporal Reinforced Swarm Optimization (ST-RSO) in which the reinforcement-guided swarm exploration, and the modeling of temporal-spatial correlations are used to select the most relevant features minimizing dimensionality and computational cost.
- The proposed deep learning-based intrusion detection model is called Multi-Scale Residual Graph Attention Network (MS-RGAN), which incorporates multi-scale graph attention and residual learning to both detect local device malfunctions and large-scale orchestrated attacks on the IoT network.
- Majority Voting Ensemble (MVE) mechanism verifies alert responses which are then validated by blockchain consensus mechanism, minimizing false alarms without tampering with logging or inaccurate incident validation.
- It includes a Distributed Trust-Aware Fault-Tolerant Defense Model (DT-FTFDM) in which the compromised devices are automatically isolated, assessed, and reintegrated securely once it passes verification. The Consensus-Validated Trust Score (CVTS) is presented in the fault validation process to make decisions regarding isolation and recovery safely.
- The general model is tested on the basis of real-world ToN-IoT data and tested on the major performance measures such as accuracy, precision, recall, F1-score, false-positive rate, processing latency, and resource efficiency and compared with the state-of-the-art intrusion detection approaches.

The structure of this paper is as follows: Chapter 1 is the introduction in which the research motivation, objectives, problem statement are presented. Section 2 analyses related literature and current methods of the Internet of Things intrusion detection. Section 3 gives a proposed blockchain-enabled MS-RGAN framework. The 4 th section is about experimental results and performance analysis. Section 5 will bring the study to a close by presenting important findings and future directions.

2. Literature review

Madi & Esteves-Verissimo (2022) aims at enhancing security and fault tolerance over container-based virtualization environments. Containers are less weighty and are less rigid yet have weaker isolation and can be attacked like container escape. The authors come up with a Fault and Intrusion Tolerance (FIT) model that employs host-level specification-based error-detection to detect violations of security by malicious containers. The framework aids in the automatic detection, recovery and fault treatment, to ensure that systems are reliable. Though its sole purpose is to address security faults, the approach can also effectively address unintentional system failures.

Mudassar et al. (2022) developed a study about fault tolerance in edge computing systems where IoT applications are to be run. The edge devices are resource-constrained and subject to failures; hence, it is difficult to ensure their reliability. The authors suggest a fault-tolerant strategy based on checkpoints and replication, according to which the IoT tasks periodically store the execution states and reproduce them on the closest edge nodes. In case of a node failure, the workload restarts at the last check point. It has been experimentally demonstrated that task reliability, availability, and deadline adherence are improved, which is why the approach can be used in latency-sensitive IoT applications.

Haseeb et al. (2022) introduces a Trust-based Fault-Tolerant Supervised Routing (Trust-FTSR) smart city IoT networks. The conventional routing techniques tend to be ineffective because links are not reliable and mobile nodes exist. The suggested model enables every node to examine the conduct of its neighbors and give trust values in order to choose trustworthy paths. Supervised machine learning algorithm increases fault tolerance without imposing

network overhead. The model enhances reliability of communication, balances energy use and prevents against attacks that use secure authentication mechanisms.

Kaur (2022) conducted a research study which investigates the fault tolerance in the Wireless Sensor Networks (WSNs). WSNs are made up of low-power nodes that are susceptible to physical breakdowns and network failure. Some fault detection and reputation-based trust models, which are employed to detect faulty nodes, are discussed in the paper. It points out the problems of manipulation of the trust and attack of confidence where the reputation systems are used by the malicious nodes. The research highlights the use of strong and flexible fault-tolerant methods to enhance the reliability of WSN.

do Nascimento & Hessel (2022) presents a decentralized framework of an intrusion detector of the IoT system based on federated learning and a distributed ledger technology. The conventional centralized IDS solutions possess scalability challenges, privacy threats and a point of failure. The suggested architecture enables IoT gadgets to jointly train intrusion detection framework without exchanging crude information. Access control is guaranteed based on blockchain and security. The method enhances privacy, scalability, and resiliency without deteriorating intrusion detection.

Sharathkumar et al. (2022) developed a fault-tolerant Software Defined Networking (SDN) architecture is proposed to recover in the prevention of DDoS attacks. SDN divides data and control planes, and thus allows flexible network management. The authors utilize Fault Tolerance (FT) methods with the help of the Shared Risk Link Group (SRLG) model and implement the stability of the network. The performance measures of delay, packet loss, recovery time are measured under attack conditions. Findings indicate that reliability in the network and recovery speed is improved during DDoS attack.

Venâncio & Duarte Jr. (2022) proposed NHAM, which is a Network Function Virtualization (NFVM) High Availability Module. NFV services should always be available to telecom carriers. NHAM allows fault tolerant virtual services with checkpointing and buffer management without any changes to VNF code. The framework advocates elastic recovery plans of stateful functions and service chain of functions. Under experimental settings, transparent fault recovery and better availability are shown in virtualized network environments.

Alghamdi & Bellaiche (2022) devoted to the improvement of intrusion detection accuracy in the IoT networks by application of ensemble learning. The current IDS models are poor in scalability and heterogeneous traffic patterns. The authors suggest a framework of Integrated Evaluation Metrics (IEM) and an automated way of ranking the best models to use in determining an ensemble classifier. Several IoT databanks are used to improve the process of detection generalization. The accuracy of the proposed approach is very high with respect to binary and multi-class intrusion detection. Comparison of Existing Fault-Tolerant and Intrusion Detection Approaches are tabulated in the following table 1.

Table 1: Comparison of Existing Fault-Tolerant and Intrusion Detection Approaches

Author (Year)	Application Domain	Core Technique Used	Strengths	Limitations
Madi & Esteves-Verissimo (2022)	Containerized environments	Specification-based fault and intrusion tolerance	Detects container escape and security breaches; automated recovery	Limited to host-level detection; no IoT-scale or learning-based detection
Mudassar et al. (2022)	Edge computing for IoT	Checkpointing and task replication	Improves reliability and task availability; meets latency deadlines	No intrusion detection; lacks security and attack awareness

Haseeb et al. (2022)	Smart city IoT networks	Trust-based supervised routing	Improves routing reliability and energy balance	Focused only on routing; no deep anomaly detection or blockchain support
Kaur (2022)	Wireless Sensor Networks	Fault detection and reputation models	Highlights fault types and trust challenges	Survey-based; no implementation or intelligent detection model
do Nascimento & Hessel (2022)	IoT intrusion detection	Federated learning + distributed ledger	Preserves privacy; decentralized detection	Limited spatio-temporal modeling; no fault recovery mechanism
Sharathkumar et al. (2022)	Software Defined Networks	Fault-tolerant SDN with SRLG	Faster recovery from DDoS attacks	SDN-specific; not applicable to heterogeneous IoT environments
Venâncio & Duarte Jr. (2022)	NFV systems	Checkpoint/restore with buffer management	Transparent fault recovery; high availability	Focused on virtual services; lacks intrusion detection intelligence
Alghamdi & Bellaiche (2022)	IoT IDS	Ensemble ML/DL with evaluation metrics	Very high detection accuracy	Centralized analysis; no blockchain or fault-tolerant recovery

2.1 Problem statement and Novelty

The fast growth of the IoT networks has revealed major threats such as compromised node, data distortion, and organized attacks such as Sybil and spoofing, which centralized IDS cannot effectively detect. IoT data is also high dimensional, is noisy and time-varying, which further reduces the accuracy of detection generating false positive and inefficient systems. This paper presents a fault-tolerant IoT intrusion detection system based on blockchain that combines the principles of secure device authentication, spatio-temporal features optimization, and anomaly detection using deep learning. It is new in that blockchain consensus is used together with the Spatio-temporal reinforced swarm optimization (ST-RSO) and Multi-Scale Residual Graph Attention Networks (MS-RGAN) to deliver decentralized, resilient, and accurate detection of advanced IoT threat.

3. Proposed model

Figure 1 shows the workflow presents an end-to-end, fault-tolerant IoT intrusion detection framework that integrates blockchain security with advanced deep learning techniques. The process begins with blockchain-based device enrolment and cryptographic authentication to ensure trusted participation and prevent spoofing or Sybil attacks. Secure data acquisition follows, where encrypted and timestamped sensor data is immutably logged on the blockchain. Preprocessing techniques such as data cleaning, normalization, and outlier removal improve data quality and learning stability. Temporal and statistical feature extraction captures device behavior and cross-node correlations, which are further refined using Spatio-Temporal Reinforced Swarm Optimization (ST-RSO) to reduce dimensionality while preserving critical dependencies. The optimized features are analysed using a Multi-Scale Residual Graph Attention Network (MS-RGAN) to detect both isolated and coordinated attacks. Alerts are validated through a Majority Voting Ensemble to reduce false positives, then confirmed using blockchain consensus. Finally, a trust-aware fault-tolerant response mechanism isolates compromised nodes and enables secure reintegration, ensuring resilience

and continuous IoT operation. Thus, the Overall architecture of the study has been schematically illustrated in figure 1.

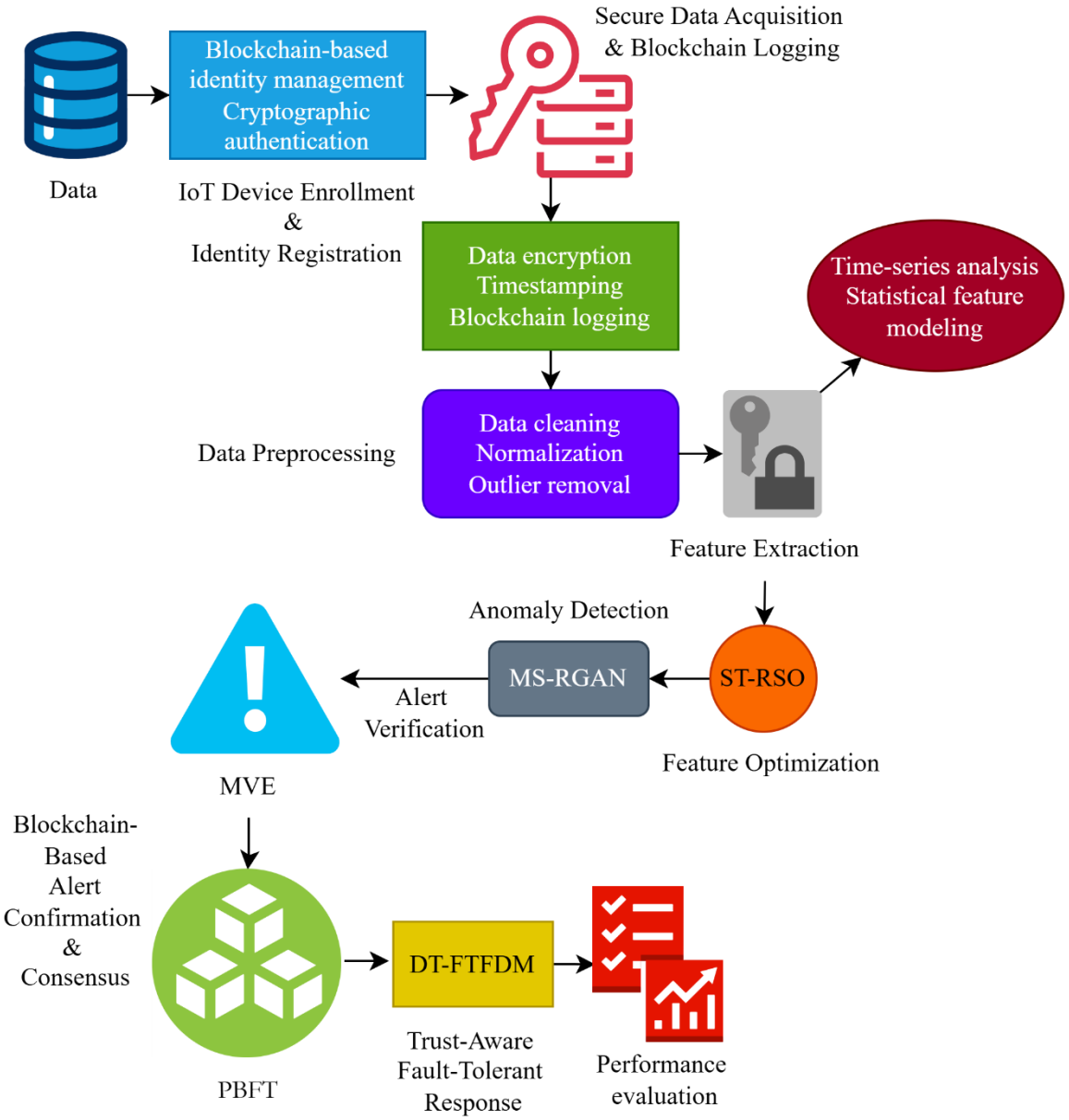


Figure 1: Overall architecture of the study

3.1 Data collection

Dataset: https://unsw-my.sharepoint.com/personal/z5025758_ad_unsw_edu_au/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fz5025758%5Fad%5Funsw%5Fedu%5Fau%2FDocuments%2FTON%5FIoT%20datasets%2FTrain%5FTest%5Fdatasets%2FTrain%5FTest%5FNetwork%5Fdataset&viewid=f8d1dec5%2Dcd5f%2D42ae%2D8b06%2D2fece580c74a&ga=1.

3.2 IoT Device Enrolment and Identity Registration

Intrusion detection in IoT networks should be fault-tolerant and this would start with the secure device enrolment and identity registration that helps to build trust among the interacting nodes. An IoT device needs to be authenticated prior to joining the network in order to eliminate attacks which are identity based, Sybil attacks, and unauthorized access. The identity management is decentralized using blockchain to avoid single points of failure and decentralizing trust. When a device is enrolled, it is allocated a distinct cryptographic identity that is authenticated and stored forever on the blockchain register. This makes the identities of the devices unchangeable, verifiable, and resistant to attack. The system provides a high level of fault tolerance in the authentication layer by ensuring its operational integrity even in the event of a maliciously acting or failed node and this is due to consensus-based validation.

Following eqn. (1) defines the cryptographic identity of the i^{th}IoT device using asymmetric key pairs. The public key is used for identity verification, while the private key ensures secure authentication and message signing.

$$ID_i = \{PK_i, SK_i\} \qquad (1)$$

Here, ID_i represents the cryptographic identity of the i^{th}IoT device, composed of a public key PK_i and a private key SK_i.

Eqn. (2) generates a unique and immutable hash for each device by combining its public key with a registration timestamp. The hashing process guarantees identity uniqueness and prevents replay or identity duplication attacks.

$$H_i = \text{Hash}(PK_i \parallel T_i) \qquad (2)$$

The hash value $H_i$is generated by hashing the public key concatenated with the registration timestamp T_i, ensuring uniqueness and temporal validity. Following eqn. (3) determines whether a device is authenticated based on the presence of its identity hash in the blockchain ledger. A valid hash confirms trusted enrolment, while absence results in access denial to maintain network security.

$$Auth_i = \begin{cases} 1, & \text{if } H_i \in B \\ 0, & \text{otherwise} \end{cases} \qquad (3)$$

Here, B denotes the blockchain ledger that stores verified device identities. The authentication flag $Auth_i$ determines whether a device is accepted into the network, where a value of 1 indicates successful authentication and 0 indicates rejection.

3.3 Secure Data Acquisition and Blockchain Logging

Upon successful authentication, the IoT devices start to collect and transfer important data in a safe manner. IoT data is very sensitive and can be easily been tampered with, this is the reason why there is encryption and blockchain logging which guarantees confidentiality, integrity and traceability. Every sensor value is encrypted and a time mark is added to it to avoid replay attacks. The data is then encrypted and stored in the blockchain, making it hard and impossible to alter and auditable. This decentralized logging scheme also makes sure that in the event of intermediate node failure or compromise, the data remains resilient and verifiable which is part of the overall fault tolerance of the intrusion detection system. The encryption of raw sensor data generated by the i^{th} IoT device at time t using a symmetric encryption function and the session key K_i ensures confidentiality of sensed data during transmission and storage, thus it can be expressed using eqn. (4),

$$C_i(t) = E_{K_i}(x_i(t)) \qquad (4)$$

where $x_i(t)$ denotes the raw sensor data generated by device i at time t. A blockchain transaction that bundles encrypted sensor data with its timestamp and device identity. The transaction structure ensures traceability, authenticity, and resistance to replay attacks and it can be expressed using eqn. (5),

$$Tx_i = \{C_i(t), T_i, ID_i\} \qquad (5)$$

The encrypted data $C_i(t)$ is produced using a symmetric encryption function $E(\cdot)$ with session key K_i. Formation of a new blockchain block by appending the hash of the validated transaction to the previous block. The chaining mechanism guarantees data immutability and tamper resistance across distributed ledger nodes and it can be given in eqn. (6),

$$B_k = B_{k\text{-}1} + \text{Hash}(Tx_i) \qquad (6)$$

Here, the blockchain block B_k is formed by appending the hash of the transaction to the previous block $B_{k\text{-}1}$, ensuring immutability and chronological consistency. The transaction Tx_i consists of the encrypted data, timestamp T_i, and device identity ID_i.

3.4 Data Preprocessing

The gathered data of the IoT should be pre-processed to enhance quality and reliability before intrusion detection. IoT data in the real world like ToN-IoT tend to be noisy, missing, imbalanced, and contain outliers which adversely affect the performance of detection. Pre-processing of data eliminates these errors by cleansing, normalizing, and removing outliers. This measure will unbiasedly guarantee the learning models are fed with clean and standardized inputs hence speeding up convergence, alleviating false alarms and fault tolerance through elimination of misleading patterns of anomaly due to corrupted data. In deep learning–based intrusion detection, normalization improves model stability and accelerates convergence, and it can be given in eqn. (7),

$$x_i^{norm} = \frac{x_i - \mu}{\sigma} \qquad (7)$$

It carries out the normalization process of raw feature values by statistically standardizing them to a common range. This operation removes mean bias and variance dominance, allowing all features to contribute equally during the learning process. Here, x_i is the original feature value, while x_i^{norm} denotes the normalized value obtained using mean μ and standard deviation σ.

The outliers are removed by checking the normalized values against a predetermined threshold value. Values that contain features that are above the threshold are substituted with the mean to discourage noise and extreme deviations. This cleaning boosts resilience and minimizes false alarms that can be brought about by abnormal and non-malicious information and it can be mathematically expressed using eqn. (8),

$$x_i^{clean} = \begin{cases} x_i^{norm}, & |x_i^{norm}| \le \theta \\ \mu, & \text{otherwise} \end{cases} \qquad (8)$$

The threshold θ defines the acceptable deviation range for detecting outliers. Values exceeding this threshold are replaced with the mean to suppress noise and extreme deviations. The resulting cleaned feature x_i^{clean} provides a stable and consistent input for subsequent feature extraction and learning stages.

3.5 Temporal and Statistical Feature Extraction

In order to accurately identify complex and coordinated attacks involving the IoT, both temporal and statistical features are derived out of the processed data. Temporal features provide time-dependent behavior and sequential patterns and allow identifying slow, stealthy, and evolving attacks. Statistics summarize the features of data distribution and cross node relationships, which are vital in the detection of abnormal traffic patterns and concerted intrusions. This integrated representation improves the capability of the system to model the local device anomalies and international network-wide deviations, and this increases the intrusion detection resilience and fault tolerance.

The mean of the feature values is computed to capture the central tendency of the i^{th} IoT device's measurements. It provides a baseline for normalization and variance calculations. This operation can be mathematically expressed using eqn. (9).

$$\mu_i = \frac{1}{N}\sum_{t=1}^{N} x_i\,(t) \qquad (9)$$

Here, $x_i(t)$ denotes the feature value of device i at time t, and Nrepresents the size of the observation window. The variance measures the spread or dispersion of the feature values around the mean for the i^{th} device. It quantifies deviations and is crucial for standardization and anomaly detection. This calculation can be mathematically expressed using eqn. (10),

$$\sigma_i^2 = \frac{1}{N}\sum_{t=1}^{N}(\,x_i(t) - \mu_i)^2 \qquad (10)$$

The parameters μ_i and σ_i^2 correspond to the mean and variance, respectively, capturing statistical dispersion. The correlation coefficient captures the linear relationship between the features of devices $i$and j, reflecting cross-node dependencies. High or low correlations indicate potential coordinated or independent behaviors in the network. This relationship can be mathematically expressed using eqn. (11),

$$Corr_{ij} = \frac{\mathrm{Cov}(x_i,x_j)}{\sigma_i\sigma_j} \qquad (11)$$

Here, correlation coefficient $Corr_{ij}$measures the dependency between devices $i$and j, where $\mathrm{Cov}(\cdot)$denotes covariance. These parameters collectively form a spatio-temporal feature vector that characterizes both individual and collective IoT behavior.

3.6 Feature Optimization using ST-RSO

After that, there comes the data preprocessing and feature engineering phase, where the features are normalized and cleaned, and finally, for identifying the most significant features, Spatio-Temporal Reinforced Swarm Optimization (ST-RSO) has been employed by this proposed model. The main goal of ST-RSO has been formulated as dimensionality reduction with emphasis on spatial as well as temporal correlations between IoT devices, which has been proven to be an important aspect in both stand-alone and multiple node attacks. The reinforced swarm agents have been trained for exploring the feature space based on rewards given for identifying importance levels of subsets related to performance optimization for such features.

Population Initialization: The first step in ST-RSO is to initialize a swarm of candidate solutions (agents). Each agent represents a subset of features randomly selected from the original feature set and it can be mathematically expressed using eqn. (12)-(13),

$$F = \{f_1, f_2, f_3,\ldots, f_n\} \qquad (12)$$

$$S_k^0 \subseteq F, k=1,2,\ldots,N_{agents} \qquad (13)$$

where $S_k^0$is the initial feature subset of agent $k$and $N_{agents}$is the total number of swarm agents. Random initialization ensures diverse coverage of the feature space and avoids premature convergence.

Fitness Evaluation: Each agent's feature subset is evaluated using a fitness function that balances anomaly detection performance and feature subset size and it is mathematically expressed using eqn. (14),

$$Fitness(S_k^t) = \alpha \cdot Accuracy(S_k^t) - \beta \cdot \frac{|S_k^t|}{n} \qquad (14)$$

Here, $Accuracy(S_k^t)$ defines the detection accuracy using the subset S_k^t, $|S_k^t|$ representing the number of selected features and α, β are the weighting parameters for balancing accuracy and dimensionality. The fitness evaluation ensures that the algorithm favors feature subsets that maximize detection performance while minimizing redundant features.

Exploration (Global Search): Exploration allows the swarm to search new regions of the feature space to avoid local optima. Agents move in the feature space influenced by the globally best solution and random perturbations and it can be expressed using eqn. (13),

$$S_k^{t+1} = S_k^t + \delta \cdot Velocity_k^t \qquad (13)$$

Here, $Velocity_k^t$ introduces stochastic changes to feature subsets and δ defines the scaling factor controlling exploration. Exploration ensures diversity and helps discover highly informative feature subsets that may not be apparent initially.

Exploitation (Local Search): Exploitation allows agents to refine feature subsets around promising solutions by incorporating reinforcement feedback and it is given in eqn. (14),

$$S_k^{t+1} = S_k^t + \gamma \cdot R_k^t \cdot (S_{best}^t - S_k^t) \qquad (14)$$

Here, R_k^t defines the reinforcement reward for agent k at iteration t, S_{best}^t is the Globally best feature subset so far and γ representing the scaling factor controlling exploitation strength. By combining exploitation with exploration, ST-RSO balances intensive local search with global feature space coverage.

Termination Criteria: The algorithm iterates until one of the following termination conditions is met:

- Maximum number of iterations T_{max} is reached.
- Improvement in fitness values between consecutive iterations falls below a threshold ϵ.
- A pre-defined detection accuracy or feature dimensionality target is achieved.

Termination ensures computational efficiency while guaranteeing high-quality feature selection.

Final Selected Features: At termination, the globally best feature subset S_{best}^{final} is selected as input for the next stage (MS-RGAN anomaly detection). This subset is compact, preserves spatial-temporal correlations, and maximizes anomaly detection accuracy and it is expressed in eqn. (15),

$$S_{best}^{final} = arg\,max_{S_k^t} \quad Fitness(S_k^t) \qquad (15)$$

The selected features reduce computational cost, improve model convergence, and enhance detection of both isolated and coordinated attacks.

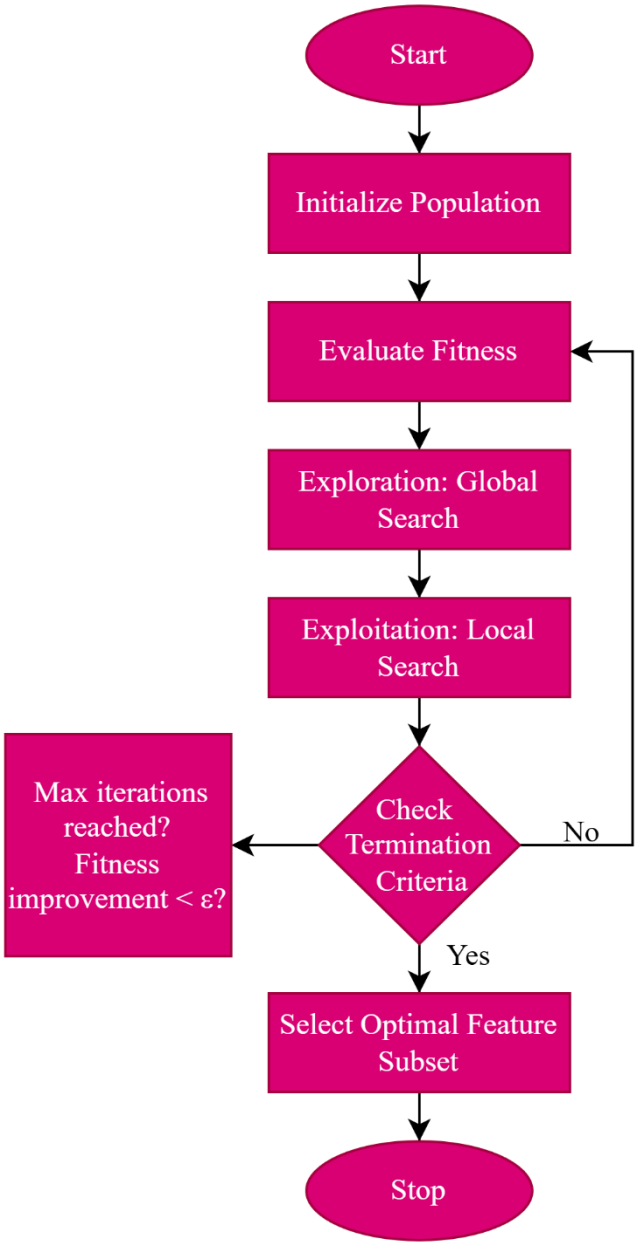


Figure 2: Flowchart for ST-RSO

Figure 2 shows the feature selection procedure of the Spatio-Temporal Reinforced Swarm Optimization (ST-RSO) algorithm accurately depicts the systematics and systemic approaches of the proposed work in identifying the optimally informative features for IoT intrusion detection systems. The system starts with the initialization of a swarm of candidate solutions, comprising feature subsets randomly selected from the original feature space. Next, there is fitness evaluation, wherein feature subsets are assessed with a function weighing the trade-off between anomaly detection capability and feature dimensionality. Agents will carry out exploration, wherein movement across feature space targets possibly valuable areas, and exploitation, wherein feature subsets are fine-tuned around the globally best solution, strengthened by rewards in a reinforcement process. This screen-to-screen action continues until the termination criterion is met, specifically after achieving the maximum number of iterations when no further

improvement in fitness is possible. On termination, the globally best feature subset is selected as the optimally selected feature for intrusion detection.

Algorithm 1: ST-RSO Feature Optimization for IoT Intrusion Detection

Input:

$F = \{f_1, f_2, f_3, ..., f_n\}$ // Original feature set after extraction

N_{agents} // Number of swarm agents

T_{max} // Maximum number of iterations

α, β // Fitness weighting parameters

γ, δ // Exploitation and exploration scaling factors

ε // Convergence threshold

Output:

S_{best}^{final} // Optimized feature subset

BEGIN

1. // Population Initialization

FOR $k = 1$ to N_{agents}

S_k^0= Random Subset (F) // Randomly select initial feature subset

END FOR

2. $t = 0$ // Initialize iteration counter

3. Evaluate Fitness for all agents:

FOR each agent k

END FOR

4. S_{best}^t = Agent with highest Fitness

5. // Iterative Optimization

WHILE $t < T_{max}$ AND $Improvement(Fitness) > \varepsilon$

FOR each agent k

// Exploration (Global Search)

$Velocity_k^t$= Random Perturbation ()

// Exploitation (Local Search)

R_k^t = Compute Reinforcement (S_k^t)

// Evaluate Fitness

// Update global best

IF Fitness (S_k^t) > Fitness (S_{best}^t)

$S_{best}^{t} = S_{k}^{t}$

END IF

END FOR

$t = t + 1$ // Increment iteration

END WHILE

6. // Select final feature subset

$S_{best}^{final} = S_{best}^{t}$

RETURN S_{best}^{final}

END

3.7 Spatio-Temporal Anomaly Detection using MS-RGAN

The set of optimal features is then given to the Multi-Scale Residual Graph Attention Network (MS-RGAN) for carrying out the anomaly detection task. In the graph attention mechanism, the attention layers represent the interactions between the devices and the global dependencies among the devices. The residual mechanism helps to improve the overall convergence of the model during the learning phase and provides optimal results even with the dynamic nature of the Internet of Things (IoT) network. The methods applied for this purpose are graph attention networks, residual learning methods, and the concept of deep learning on multiple scales.

Node Feature Embedding: Each IoT device i is represented as a node in a graph. Its feature vector $h_i$encodes device behavior from the optimized features and it is given in eqn. (16),

$$h_i \in \mathbb{R}^d \qquad (16)$$

Here, d is the feature dimension after ST-RSO and h_i is the node embedding representing device i.

Graph Attention Layer: Node embeddings are updated based on neighbors N_i using attention weights α_{ij} to emphasize critical interactions and it is given in eqn. (17),

$$h_i^{'} = \sigma(\sum_{j \in N_i} \alpha_{ij} W h_j) \qquad (17)$$

Here, $h_i^{'}$ is the updated embedding of node I, h_j defines the feature vector of neighboring node j, W is the learnable transformation matrix, σ states the activation function (e.g., ReLU, LeakyReLU) and α_{ij} is the attention weight for the edge (i, j).

Temporal-Spatial Attention Parameter: A novel parameter τ_{ij}encodes temporal correlations between nodes i and j across time, improving detection of coordinated or evolving attacks. Attention weights are computed using eqn. (18),

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(a^T[h_i||h_j||\tau_{ij}]))}{\sum_{k \in N_i} \exp(\text{LeakyReLU}(a^T[h_i||h_k||\tau_{ik}]))} \qquad (18)$$

Here, a is the learnable attention vector, $||$ defines the concatenation operator and τ_{ij} is the temporal-spatial attention parameter capturing evolving interactions.

Multi-Scale Feature Aggregation: MS-RGAN uses multi-scale aggregation to capture anomalies at different granularities and it is given in eqn. (19),

$$H^{(l+1)}=\text{Concat}(H_s^{(l)}, H_g^{(l)}) + H^{(l)} \qquad (19)$$

Here, $H_s^{(l)}$ is the local-scale aggregation (single-node deviations), $H_g^{(l)}$ is the global-scale aggregation (coordinated attacks) and Residual connection $+H^{(l)}$improves training stability. Then, after multi-scale aggregation and the attention-based update of embeddings, finally, the node embedding $h_i^{'}$ captures the local and global properties of device i in space and time. These are then used in a classifier layer to calculate probabilities for whether or not a node is normal/anomaly and it is given in eqn. (20),

$$y_i=\text{Softmax}(Uh_i^{'}) \qquad (20)$$

Where, $h_i^{'} \in \mathbb{R}^d$is the final embedding of node $i$after MS-RGAN layers, $U \in \mathbb{R}^{2\times d}$is the learnable weight matrix for mapping embeddings to two output classes (normal/anomalous) and $y_i \in \mathbb{R}^2$is the probability vector for the two classes. Thus, it can be expressed using eqn. (21),

$$y_i = \begin{bmatrix} p(\text{Normal}) \\ p(\text{Anomalous}) \end{bmatrix}, p(\text{Normal}) + p(\text{Anomalous})=1 \qquad (21)$$

The final classification is determined by the maximum probability and it is given in eqn. (22),

$$\text{Class}_i = \begin{Bmatrix} \text{Normal} & \text{if } p(\text{Normal}) > p(\text{Anomalous}) \\ \text{Anomalous} & \text{otherwise} \end{Bmatrix} \qquad (22)$$

Here, the softmax ensures normalized probabilities, while the argmax operation assigns the final class. This approach allows MS-RGAN to continuously monitor IoT devices, distinguishing isolated anomalies from coordinated attacks efficiently.

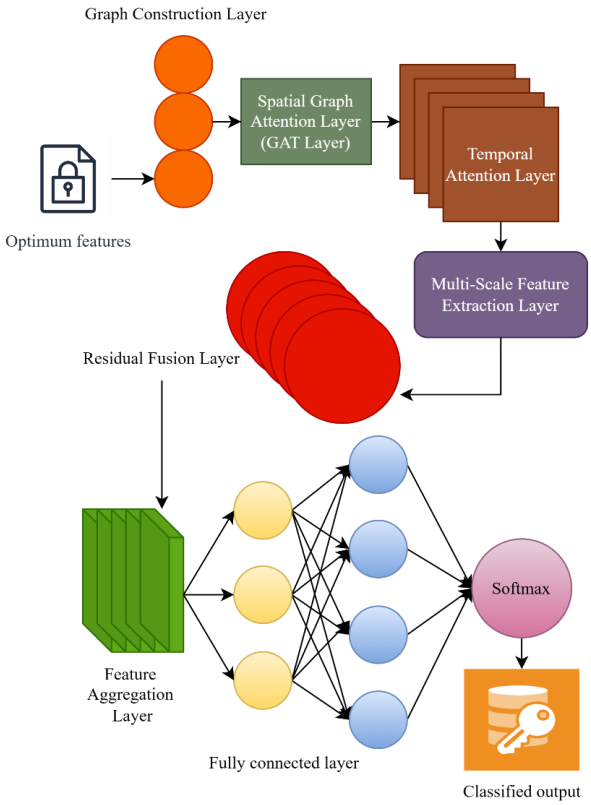


Figure 3: Architecture of MS-RGAN for Anomaly Detection

Figure 3 shows the architecture for MS-RGAN and it is used to perform spatiotemporal anomaly detection in the IoT networks by identifying the collective actions of the devices as well as the evolution of the patterns. Optimized features are first transformed into a dynamic graph, where the nodes are considered as the IoT devices and the edges are the communication associations among them. Spatial graph attention layers are used to apply weighted emphasis to the neighboring devices based on the most significant associations that are generally connected with anomalies. On the other hand, the temporal attention focuses on the variations with time that introduce correlations among the devices based on the evolution of the patterns. Additionally, the outcomes are transformed from the multi-scale feature learning process using residual connections that enhance the stability of the training as well as the convergence speed. Ultimately, the classification module is applied to determine the anomalies of the devices as well as the massive networks.

3.8 Alert Verification via Majority Voting Ensemble (MVE)

After the possible anomalies are identified by MS-RGAN, the system uses the Majority Voting Ensemble (MVE) method in validating alerts. This process involves combining the opinions or predictions of several classification models, which will activate the final alert if the majority concur. This serves to minimize false alerts, which are less significant prior to carrying out any actions related to security. This method utilizes both the ideas of combining several classification models through ensemble learning, as well as majority voting. The mathematical formulation of the MVE is given in eqn. (23),

$$y_i^{\text{MVE}}=\text{mode}\{y_i^{(1)}, y_i^{(2)}, \dots, y_i^{(m)}\} \quad (23)$$

Here, $y_i^{(k)} \in \{0,1\}$ is the prediction of the k^{th}classifier for node i, with 0 = Normal, 1 = Anomalous, m is the total number of classifiers in the ensemble and $\text{mode}\{\cdot\}$ computes the most frequent prediction among the classifiers. Equivalently, the decision can be expressed as a thresholded sum and it is given in eqn. (24),

$$y_i^{\text{MVE}} = \begin{Bmatrix} 1 & \text{if } \sum_{k=1}^{m} y_i^{(k)} \geq \lceil m/2 \rceil \\ 0 & \text{otherwise} \end{Bmatrix} \quad (24)$$

Here, $\lceil m/2 \rceil$ is the majority threshold, If the sum of anomaly votes reaches or exceeds the threshold, the node is classified as anomalous. Otherwise, it is considered normal. y_i^{MVE} is the final ensemble-based alert for node i, $y_i^{(k)}$ defines the individual classifier predictions, m representing the number of classifiers participating in majority voting. With this ensemble verification, the effects of classification mistakes made in a single model can be mitigated in a manner that ensures fault tolerance and high-confidence alert validation in a distributed IoT environment.

3.9 Blockchain-Based Alert Confirmation and Consensus

Following validation by Majority Voting Ensemble (MVE), a verified alert is then stored on a blockchain so that it is immutable and traceable. An alert transaction is then encrypted, timestamped, and signed and then recorded on the blockchain and it can be given in eqn. (25)-(26),

$$Tx_i = \{C_i(t), T_i, ID_i\} \quad (25)$$

$$B_k = B_{k\text{-}1} + \text{Hash}(Tx_i) \quad (26)$$

Here, Tx_i is the transaction representing alert from node i at time t, $C_i(t)$ is the anomaly classification (from MVE) for node i, T_i is the timestamp of alert generation, ID_i defines the unique device identifier, B_k is the blockchain state after adding transaction Tx_i and $Hash(\cdot)$is the cryptographic hash function ensuring immutability. The blockchain employs consensus mechanism, such as PBFT (Practical Byzantine Fault Tolerance), to achieve decentralized validation and it is given in eqn. (27),

$$B_k = \text{Consensus}(B_{k\text{-}1}, Tx_i) \quad (27)$$

This ensures that most nodes agree on the inclusion of the transaction. Decentralized Consensus removes the single point of failure, providing fault-tolerant logging of all verified alerts. Each party in the network holds a secure and timestamped record of all alerts.

3.10 Trust-Aware Fault-Tolerant Response (DT-FTFDM)

After the alerts are securely recorded, the DT-FTFDM will respond to the threats. The infected nodes are isolated to stop the attack diffusion process. After the alerts are securely recorded, the DT-FTFDM will respond to the threats. The infected nodes are isolated to stop the attack diffusion process. Thus, it can be mathematically evaluated using eqn. (28),

$$CVTS_i(t+1) = \gamma \cdot CVTS_i(t) + (1\text{-}\gamma) \cdot R_i(t) \quad (28)$$

Where, $CVTS_i(t)$ is the consensus-Validated Trust Score of node i at time t, $R_i(t)$ is the observed reliability/recovery feedback from network consensus and γ defines the trust decay factor, balancing historical trust vs. recent behavior. Nodes with $CVTS_i(t+1) < \theta_{\text{trust}}$ are automatically quarantined, while nodes with $CVTS_i(t+1) \geq \theta_{\text{trust}}$ are reintegration-eligible after verification. The final node recovery decision is formalized in eqn. (29),

$$\text{NodeStatus}_i = \begin{cases} \text{Isolated} & CVTS_i < \theta_{\text{trust}} \\ \text{Active} & CVTS_i \geq \theta_{\text{trust}} \end{cases} \quad (29)$$

Where, $CVTS_i$ is the dynamic trust score of device i, used for fault-tolerant recovery, θ_{trust} is the threshold to determine isolation or reintegration, γ is the weight for previous trust history, controlling adaptation speed, $R_i(t)$ defines the reliability feedback or consensus-based evaluation for node i, and NodeStatus_i is the current operational status of node i. This combination of blockchain-based consensus and trust-aware fault tolerant response ensures resilience, security, and operability of IoT network even under partial nodes failures or coordinated attacks. With continuous update of CVTS and Tamper-proof logs, dynamic, adaptive, and end-to-end fault-tolerant intrusion detection is achieved.

4. Result analysis

This article introduces the detailed performance assessment of the proposed MS-RGAN-based intrusion detection system enabled with the support of the blockchain technology. A comparison study is conducted among the proposed system and other pre-eminent machine learning and deep learning models, Random Forest, SVM, CNN-BiLSTM, and GCN, using various parameters. Critical parameters such as accuracy, precision, recall, F1 score, FP rate, FN rate, MCC, detection time, reduction ratio of features, and consensus verification time are measured and examined comprehensively. These findings authenticate the efficiency and effectiveness of the proposed technique to maximize the detection accuracy, suppress the FP rate, and eliminate the detection and verification time overheads, thereby justifying the appropriateness of the proposed technique to ensure the fault-tolerant security of the IoT networks.

4.1 Experimental setup

Table 2 summarizes the dataset configuration, model architecture, training hyperparameters, optimization strategies, and system settings used to evaluate the proposed spatio-temporal anomaly detection framework.

Table 2: Experimental Setup and Configuration Parameters

Category	Parameter	Description
Dataset	IoT Traffic Dataset	UNSW-NB15 / TON-IoT / Custom IoT dataset

Data Split	Training: Validation: Testing	70%: 15%: 15%
Pre-processing	Normalization	Min–Max normalization
	Missing value handling	Mean imputation
Feature Optimization	Optimization Algorithm	ST-RSO
	Swarm size	30 agents
	Max iterations	100
	Reinforcement reward	Detection gain − feature penalty
	Selected features	Adaptive (reduced dimensionality)
Graph Construction	Nodes	IoT devices
	Edges	Communication / proximity based
MS-RGAN Architecture	Input layer	Optimized feature vectors
	Graph Attention Layers	2–3 GAT layers
	Attention heads	4 heads per layer
	Temporal Attention Layer	Enabled
	Multi-Scale Modules	Short-term and long-term windows
	Residual Connections	Yes
Training Parameters	Batch size	8
	Learning rate	0.001
	Optimizer	Adam
	Loss function	Categorical Cross-Entropy
	Activation function	ReLU, Softmax (output)
	Epochs	50–100
Anomaly Detection	Classifier	Softmax layer
	Output classes	Normal / Anomalous
Alert Verification	Ensemble models	SVM, Random Forest, MS-RGAN
	Voting method	Majority Voting Ensemble (MVE)
Blockchain Setup	Consensus protocol	PBFT / Raft
	Block contents	Alert ID, timestamp, hash
	Ledger type	Permissioned blockchain
Response Mechanism	Defense model	DT-FTFDM
	Trust score	Consensus-Validated Trust Score (CVTS)
	Isolation mechanism	Automatic node quarantine
Evaluation Metrics	Performance metrics	Accuracy, Precision, Recall, F1-Score
	Security metrics	False Alarm Rate, Detection Rate
Hardware Setup	Processor	Intel i7 / Xeon
	RAM	16–32 GB

	GPU	NVIDIA RTX / Tesla (optional)
Software Environment	Framework	Python, PyTorch / TensorFlow
	OS	Ubuntu / Windows

4.2 Metrics analysis

Table 3 provides an extensive analysis of the proposed model based on classification accuracy, error rate, reliability, efficiency, and optimization. The mentioned parameters above help in calculating the correctness of detection, rate of false alarm occurrences, efficiency of dimensionality reduction, delay in computation, as well as overhead of consensus validation. These parameters help in performing an extensive analysis of the proposed model based on its efficiency and accuracy of intruder detection.

Table 3: Analysis on performance metrics

Metric	Description	Mathematical expression
Accuracy (%)	Overall correctness of the model in classifying both normal and attack samples	$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100$
Precision (%)	Proportion of correctly detected attacks among all predicted attacks	$Precision = \frac{TP}{TP + FP} \times 100$
Recall (%)	Ability of the model to correctly detect actual attacks	$Recall = \frac{TP}{TP + FN} \times 100$
F1-Score (%)	Harmonic mean of precision and recall, balancing detection quality	$F1{=}2 \times \frac{Precision \times Recall}{Precision + Recall}$
False Positive Rate (FPR) (%)	Percentage of normal nodes incorrectly classified as attacks	$FPR = \frac{FP}{FP + TN} \times 100$
False Negative Rate (FNR) (%)	Percentage of attacks incorrectly classified as normal	$FNR = \frac{FN}{FN + TP} \times 100$
Negative Predictive Value (NPV) (%)	Probability that a predicted normal node is truly normal	$NPV = \frac{TN}{TN + FN} \times 100$
Matthews Correlation Coefficient (MCC)	Balanced measure considering all confusion matrix values	$MCC = \frac{(TP \cdot TN - FP \cdot FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$
Detection Latency (ms)	Time taken to detect an anomaly after data arrival	$Latency = t_{detect} - t_{arrival}$
Feature Reduction Ratio (%)	Percentage reduction in features after ST-RSO optimization	$FRR(\%) = \left(1 - \frac{F_{\text{opt}}}{F_{\text{orig}}}\right) \times 100$

Consensus Validation Time (ms)	Time required to validate alerts using MVE + Blockchain consensus	$CVT = t_{consensus_end} - t_{consensus_start}$

4.3 Performance analysis

These proposed models include a wide range of progressive learning complexities for a fair comparison to be made. The Random Forest and SVM models have been selected because of their popularity and ability to handle problems such as intrusion detection. The CNN-BiLSTM model has been selected because it is able to process space and temporal dependencies for the sequence data formed by the IoT networks. The GCN model has been selected because it is able to capture the graph structure of the IoT networks along with the dependencies for the other devices. The MS-RGAN model has been selected because it is able to handle problems such as anomaly detection, low latency, and fault tolerance. Table 4 shows the Comparison of performance analysis.

Table 4: Comparison of performance analysis

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	FPR (%)	FNR (%)	NPV (%)	MCC	Detection Latency (ms)	Feature reduction Ratio (%)	Consensus Validation Time (ms)
Random Forest [21]	90.4	89.7	89.9	89.8	4.8	5.6	88.9	79	42	12	28.6
SVM [22]	91.6	91.1	91.3	91.2	4.1	4.7	90.6	82	39	15	26.9
CNN-BiLSTM [23]	94.8	94.4	94.6	94.5	2.9	3.4	94	88	31	18.5	22.4
GCN [24]	95.6	95.2	95.4	95.3	2.4	3	94.9	90	29	22.3	20.7
Proposed MS-RGAN	98.7	98.4	98.6	98.5	0.9	1.2	98.2	96	21	41.8	14.2

From the numerical analysis, it can be noted that there is a continuous improvement in performance with the increased expressiveness of the modeling paradigm. Random Forest classifier reports an accuracy of 90.4% with FPR = 4.8% and FNR = 5.6% with moderate classification error in realistic NF attack cases. SVM slightly increments the accuracy measure to 91.6% with decreased FPR = 4.1% and FNR = 4.7% but with relatively high latency (39 ms) and modest feature extraction (15%). CNN-BiLSTM exhibits substantial increments in the accuracy measure to 94.8%, with decreased FPR = 2.9% and FNR = 3.4%, thus validating the effectiveness of learning both space and temporal features together. The GCN reports further improvements in the accuracy measure to 95.6%, with decreased latency (29 ms) and increased MCC (MCC = 90), thus validating the usefulness of inter-device relationships for improving detection performance.

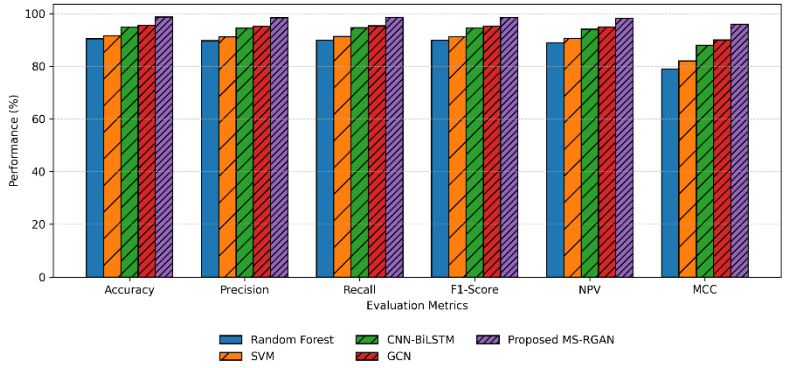


Figure 4: Graphical representation of performance comparison

With a focus on accuracy, the best performers are found to be MS-RGAN, SDRRGAN, and SRGAN, achieving an accuracy of 98.7%, precision of 98.4%, recall of 98.6%, and F1 score of 98.5%, along with a drastically reduced FPR of 0.9% and FNR of 1.2%. With a very high level of balanced prediction, MS-RGAN has an MCC of 96. Additionally, it shows the lowest detection latency of 21 ms, consensus validation time of 14.2 ms, and highest feature reduction ratio of 41.8%, emphasizing not only accuracy but also efficiency. These results clearly impose that multi-scale graph attention with residual learning is very efficient in learning dynamic attack patterns to provide fast, accurate, and reliable intrusion detection to dynamic IoT scenarios. Figure 4 shows the graphical representation of performance comparison.

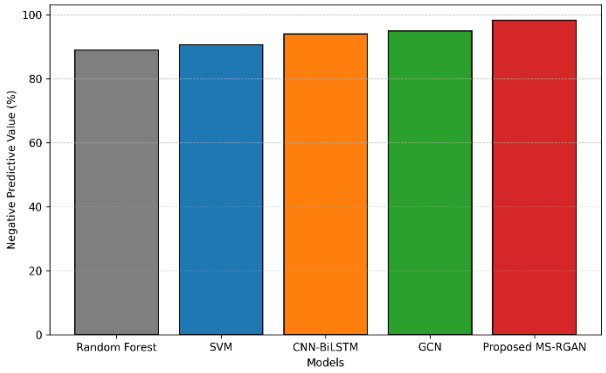
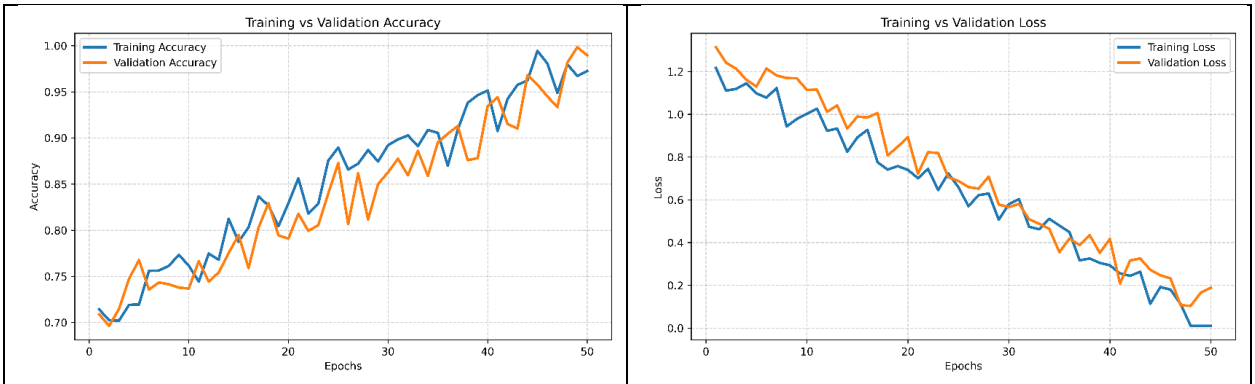


Figure 5: Analysis on NPV

Figure 5 shows the comparison of Negative Predictive Value (NPV) gradually enhances in all the models under analysis, resulting in remarkable certainty in discerning normal traffic. The NPV measured in the Random Forest model is 88.9%, which enhances to 90.6% in SVM, signifying less false negativity. The CNN-BiLSTM model again enhances the NPV to 94% by aptly recognizing time-series dependencies. The GCN model measures 94.9%, leveraging graph representation. The proposed MS-RGAN models attain the highest NPV of 98.2%, signifying stronger confidence in measuring normal class with fault tolerance in intrusion detection.



Training vs testing accuracy	Training vs testing loss

Figure 6 (a)-(b): Training vs Testing (a) Accuracy (b) Loss

The training and validation accuracy curves in figure 6 (a) show steady improvement over 50 epochs, rising from ~0.70 to over 0.98. Both curves closely track each other with minimal gap, indicating effective learning, good generalization, and no significant overfitting in the MS-RGAN model. Both training and validation losses in figure 6 (b) shows decreases consistently from ~1.2 to below 0.2 over 50 epochs, showing effective optimization. The curves

remain closely aligned with only minor divergence, confirming stable convergence, strong generalization, and minimal overfitting in the MS-RGAN model.

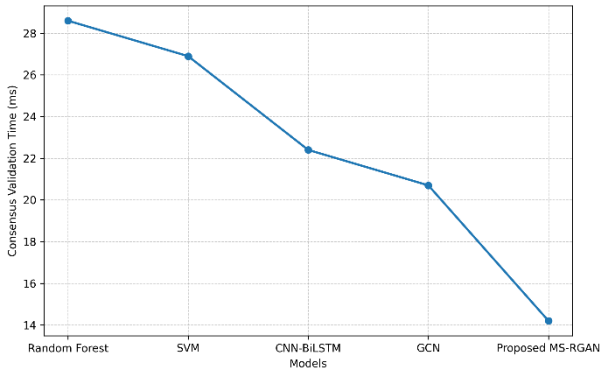


Figure 7: Analysis on Consensus validation time (ms)

Figure 7 shows the consensus validation time and this highlights the efficiency of MS-RGAN in blockchain-enabled IoT intrusion detection. Random Forest and SVM require 28.6 ms and 26.9 ms, while CNN-BiLSTM and GCN reduce it to 22.4 ms and 20.7 ms. MS-RGAN achieves the fastest validation at 14.2 ms, ensuring rapid and secure data consensus.

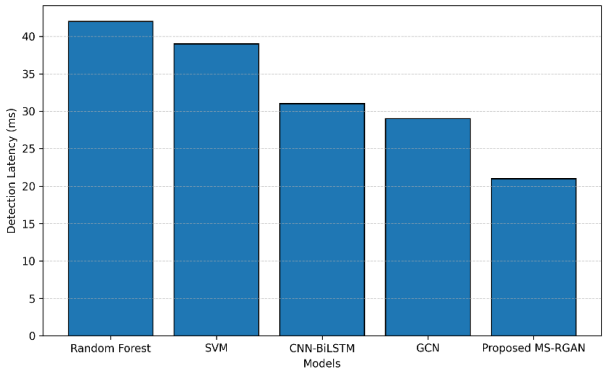


Figure 8: Analysis on detection latency (ms)

Figure 8 shows the detection latency and it shows that the detection latency is significantly improved compared to conventional models. Random Forest and SVM result in latencies of 42 ms and 39 ms, while for CNN-BiLSTM and GCN, these are further reduced to 31 ms and 29 ms, respectively. Notably, with the proposed MS-RGAN, it reaches the minimum with 21 ms latency, proving that it could perform the real-time detection with higher efficiency. These values testify that MS-RGAN efficiently processes spatio-temporal IoT data, thus offering faster intrusion response and fault-tolerant security.

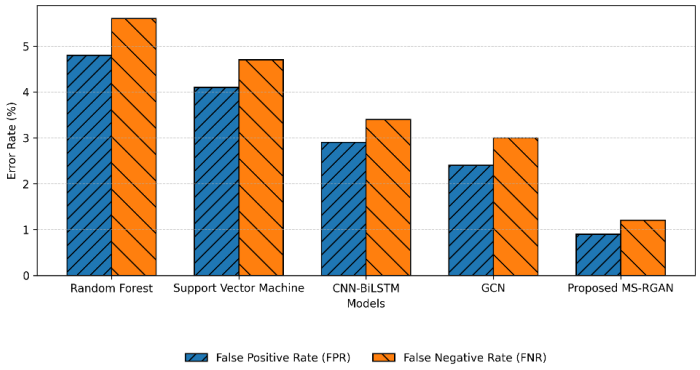


Figure 9: Comparison of Error rate

The comparison of the False Positive Rate and False Negative Rate is represented through figure 9, which shows the clear decline in the more complex models. The error rates of the Random Forest and SVM models are relatively high, with FPR and FNR above 4% and 4.5%, respectively. The CNN-BiLSTM model shows an impressive decline in these error rates, and the GCN further decreases the error rates to 2.4% and 3% for FPR and FNR, respectively, using the properties of the graph. The proposed MS-RGAN shows the lowest error rates of 0.9% and 1.2% for FPR and FNR, respectively.

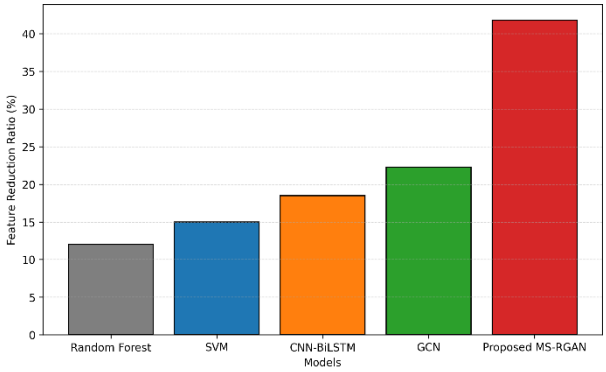


Figure 10: Comparison of feature reduction ratio

Fig. 10 compares the feature reduction ratio, which improves consistently with the models, indicating their superior capability of selecting features. Random Forest and SVM yield a moderate result with a reduction ratio of 12% and 15%, respectively. By learning deeper features, CNN-BiLSTM and GCN improve the reduction ratio to 18.5% and 22.3%, respectively. However, the proposed MS-RGAN achieves a much higher reduction ratio of 41.8%, exhibiting outstanding effectiveness at retaining the most discriminative features with reduced complexity.

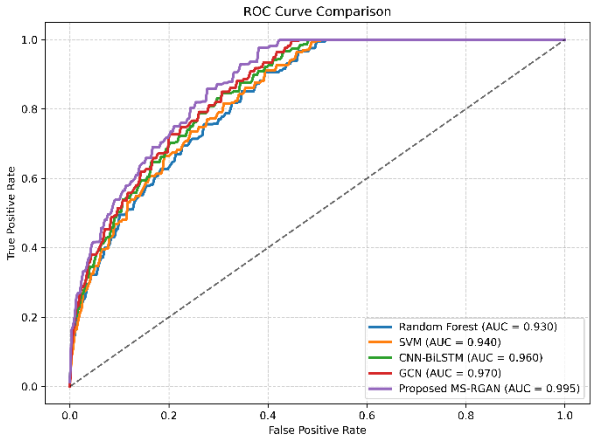
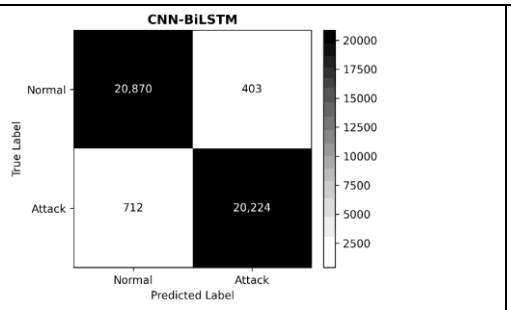
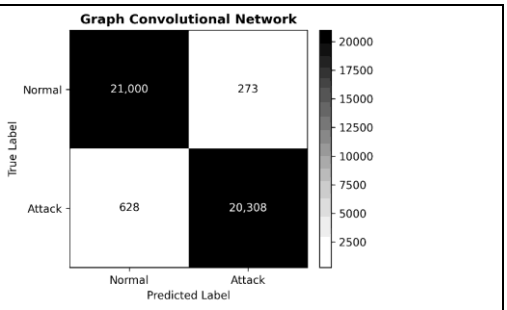


Figure 11: ROC analysis

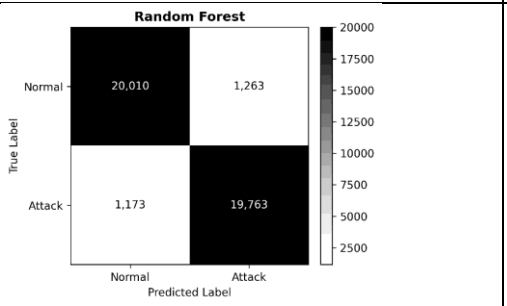
A ROC curve analysis of various models is shown in figure 11. This clearly reveals the efficacy and superiority of our proposed MS-RGAN model over others for developing a blockchain-enabled fault-tolerant IoT intrusion detection system. Our MS-RGAN has evidently recorded an AUC of 0.995. This has outperformed other models includingRandom Forest AUC of 0.930, SVM AUC of 0.940, CNN-BiLSTM AUC of 0.960, and GCN AUC of 0.970. This graph reveals that our MS-RGAN has been able to distinguish between normal and malicious network traffic with an impeccable level of accuracy. This has proved its efficiency and efficacy for modeling intricate spatio-temporal attributes of IoT network flows and a blockchain-based decentralized and tamper-proof alert handling approach. The AUC value of 0.995 has clearly proved the model's accuracy and efficiency for detecting complex attacks, reduced frequencies of false triggers of alerts, and its adeptness in coping within IoT environments. This has clearly established its efficacy for developing an intrusion detecting system.



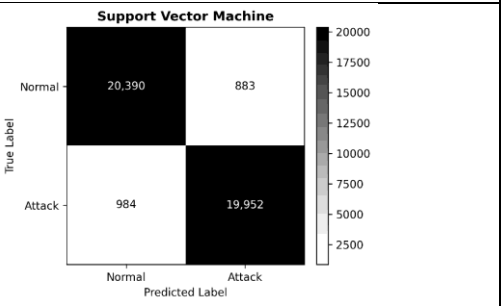
CNN-BiLSTM confusion matrix



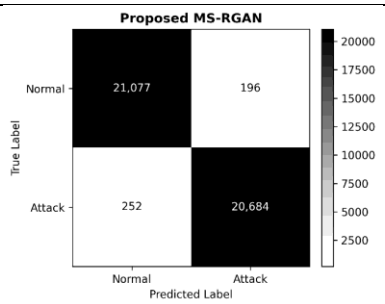
GCN confusion matrix



Random forest confusion matrix



SVM confusion matrix



(e) Confusion matrix for proposed model

Figure 12 (a)-(e): Analysis on confusion matrix

The confusion matrix of CNN-BiLSTM in Figure 12(a) presents the strong performance-genuine prediction of normal and attack instances is 20,870 and 20,224 respectively with only 1,115 misclassifications. In Figure 12(b), the confusion matrix of GCN model presents that it discriminates normal and attack records very accurately in which it predicts 21000 instances as normal and 20308 as attack instances with only 901 misclassifications. The confusion matrix for a random forest is shown in Figure 12(c) and it presents very high discriminatory power by classifying 20010 instances as true normals and 19763 instances as true attacks while misclassifying 2436 instances. Figure 12(d) shows the confusion matrix for the SVM model and it classifies 20390 normal and 19952 attack instances correctly and resulted in only 1867 misclassifications, hence presenting outstanding detection performance. Conclusively, proposed MS-RGAN model outperforms the state-of-the-art anomaly detection models presented in Figure 12(e) where it successfully predicted 21077 normal and 20684 attack instances with only 448 misclassifications.

5. Discussion

The proposed blockchain-based fault tolerant IoT intrusion detection framework has shown considerable improvements in ensuring high detection accuracy by means of MS-RGAN, low false positives by means of MVE, and tamper-proof decision logging by means of PBFT/Raft consensus. The proposed framework tackles node attacks, Sybil attacks, and integrity attacks. However, the major limitations include the significant computational and storage complexity of the blockchain operations as well as MS-RGAN. The real-time nature might be difficult to achieve because the consensus mechanism may cause latency in terms of energy consumption. Future research should, therefore, focus on exploring lightweight versions of the blockchain technology.

6. Conclusion

In this paper, an innovative, blockchain-based, and fault-tolerant IoT intrusion detection system framework utilizing the MS-RGAN spatio-temporal deep learning model is proposed. The proposed system framework consists of several steps, including IoT network traffic retrievals from multiple devices, data pre-processing, data normalization, and noise elimination. A feature extraction and dimensionality reduction technique is further applied for the elimination of redundant features and to boost the efficiency of learning. The pre-processed features are then fed into the proposed MS-RGAN system, wherein the multi-scale representation technique captures the corresponding spatial and temporal relationships of intrusions. The generative adversarial learning method further refines the intrusion class boundaries and improves their resilience against class imbalance problems. Finally, the results of intrusion detection are securely validated and recorded using an efficient and fast blockchain-based consensus algorithm, and finally, real-time intrusion alerts are produced with negligible latency. The proposed MS-RGAN model demonstrates dramatic improvements in performance over current models, with improvements in accuracy ranging from 3.2% to 9.2%, precision/recall/F1 from 3.4% to 9.7%, and MCC from 6.7% to 21.5%, with decreased false positives and negatives by 60-81%, detection latency by 27.6-50%, and a shorter consensus time by 31.4-50.3%, as well as improved feature reduction by 87-248%. The proposed integrated system consisting of MS-RGAN and

blockchain, hence, presents an efficient and scalable next-generation solution for real-world IoT intrusion detection tasks.

6.1 Future scope

The blockchain-supported MS-RGAN intrusion detection system proposed in this paper aims to be an extended research work towards addressing the newly rising security issues in the future. The research aims to incorporate federated learning into the current model, which will help perform collaborative learning within different domains for a secure IoT environment. The incorporation of online learning mechanisms will allow addressing the rising attack patterns in real-time. The blockchain component can thus be optimized with energy-effective consensus algorithms for large-scale environments. The proposed model can also be extended for diverse IoT communication environments and collaborate with edge and cloud environments for real-world applicability. The proposed model can also incorporate Explainable AI for increased transparency and confidence in intrusion detection decisions.

REFERENCES

1. Bhatkar, A. G., Gupta, S., & Patel, P. (2025). ST-IDS: Spatio-Temporal Feature-Based Multi-Tier Intrusion Detection System for Artificial Intelligence-Powered Connected Autonomous Vehicles. *Expert Systems*, *42*(5), e70026.
2. Yang, X., Cui, Y., & Li, H. (2025). M3NID: an intrusion detection system based on the dual-mode spatio-temporal feature fusion method. *PeerJ Computer Science*, *11*, e3393.
3. Canonico, R., Lista, F., Navarro, A., Sperlí, G., & Vignali, A. (2025). Threat detection in reconfigurable Cyber-Physical Systems through Spatio-Temporal Anomaly Detection using graph attention network. *Computers & Security*, 104509.
4. Qin, G., & Liang, Y. Intrusion Detection System for Connected Autonomous Vehicles Using Spatio-Temporal Information.
5. Ishtiaq, W., Zannat, A., Parvez, A. S., Hossain, M. A., Kanchan, M. H., & Tarek, M. M. (2025). CST-AFNet: A dual attention-based deep learning framework for intrusion detection in IoT networks. *Array*, 100501.
6. Wang, Y., Wang, X., Yi, B., Huang, M., & Kou, Y. (2025). P3Fed: A Personalized and Privacy-Preserving Federated Framework for Intrusion Detection in Computing Power Network. *Computer Networks*, 111886.
7. Yang, Z., Zhang, J., & Su, B. (2025). Internet of vehicles intrusion detection method based on CFS-COA feature selection and spatio-temporal feature extraction. *The Computer Journal*, *68*(5), 487-501.
8. Khan, S., Siddique, T. H. M., Ibrahim, M. S., Siddiqui, A. J., & Huang, K. (2025). Spatio-temporal deep learning for improved face presentation attack detection. *Knowledge-Based Systems*, *311*, 113059.
9. Ding, P., Chen, H., Yang, X., & Yang, Q. (2025). Intrusion Detection Method for Power Internet of Things Based on Data Hybrid Detection Model Considering the Spatio-temporal Characteristics. *International Journal of High Speed Electronics and Systems*, 2540437.
10. Benahmed, H., M'hamedi, M., Merzoug, M., Hadjila, M., Bekkouche, A., Etchiali, A., & Mahmoudi, S. (2025). HBiLD-IDS: An Efficient Hybrid BiLSTM-DNN Model for Real-Time Intrusion Detection in IoMT Networks. *Information*, *16*(8), 669.
11. Prajwalasimha, S. N., & Sumathi, D. (2025, August). Event-Driven Intrusion Detection Systems using Spiking Neural Networks for Edge and IoT Security. In *2025 5th International Conference on Soft Computing for Security Applications (ICSCSA)* (pp. 41-47). IEEE.
12. Farrukh, Y. A., Wali, S., Khan, I., & Bastian, N. D. (2025). Xg-nid: Dual-modality network intrusion detection using a heterogeneous graph neural network and large language model. *Expert Systems with Applications*, 128089.
13. Madi, T., & Esteves-Verissimo, P. (2022, September). A fault and intrusion tolerance framework for containerized environments: A specification-based error detection approach. In *2022 International Workshop on Secure and Reliable Microservices and Containers (SRMC)* (pp. 1-8). IEEE.
14. Mudassar, M., Zhai, Y., & Lejian, L. (2022). Adaptive fault-tolerant strategy for latency-aware IoT application executing in edge computing environment. *IEEE Internet of Things Journal*, *9*(15), 13250-13262.
15. Haseeb, K., Saba, T., Rehman, A., Ahmed, Z., Song, H. H., & Wang, H. H. (2022). Trust management with fault-tolerant supervised routing for smart cities using internet of things. *IEEE internet of things journal*, *9*(22), 22608-22617.
16. Kaur, T. (2022). Analyze and evaluate the state of art scheme for fault tolerant WSN. *International Journal of Research in Informative Science Application & Techniques (IJRISAT)*, *6*(10-12), 1-12.
17. do Nascimento, F. A. M., & Hessel, F. (2022, March). A decentralized federated learning architecture for intrusion detection in IoT systems. In *International Conference on Advanced Information Networking and Applications* (pp. 256-268). Cham: Springer International Publishing.
18. Sharathkumar, S., Paul, A. J., & Sreenath, N. (2022, December). Design of Reliable Fault Tolerant Architecture and Analysis of a Software Defined Network to recover from DDoS Attack. In *2022 IEEE 7th International Conference on Recent Advances and Innovations in Engineering (ICRAIE)* (Vol. 7, pp. 206-213). IEEE.

19. Venâncio, G., & Duarte Jr, E. P. (2022, November). NHAM: An nfv high availability architecture for building fault-tolerant stateful virtual functions and services. In *Proceedings of the 11th Latin-American Symposium on Dependable Computing* (pp. 35-44).
20. Alghamdi, R., & Bellaiche, M. (2022). Evaluation and selection models for ensemble intrusion detection systems in IoT. *IoT*, *3*(2), 285-314.
21. Al-hubaishi, M., & Abdulraqeb, M. (2025). Hybrid CNN-LSTM Model with Random Forest Classifier for Intrusion Detection in Connected Vehicles. *International Journal of Automotive Science And Technology*, *9*(4), 686-696.
22. Ridwan, L. (2025). Comparative Analysis of Hybrid AI Models (CNN–RNN, Autoencoder–SVM) for Network Intrusion Detection.
23. Al Amin, M., Ahsan, M. S., Shoaib, H. A., Eva, A. A., Sayeema, A., & Firdaus, S. (2025, July). Explainable Deep Malware Detection in IoT Devices Using CNN-BiLSTM. In *2025 International Conference on Quantum Photonics, Artificial Intelligence, and Networking (QPAIN)* (pp. 1-6). IEEE.
24. Baniata, L. H., ALDabbas, A., Atwan, J. M., Alahmer, H., Elmasri, B., & Bunterngchit, C. (2025). A Dual-Attention CNN–GCN–BiLSTM Framework for Intelligent Intrusion Detection in Wireless Sensor Networks. *Future Internet*, *18*(1), 5.