Reinforcement learning-based intrusion prevention in M2M systems

Abstract

The increasing reliance on Machine-to-Machine (M2M) communication in smart environments has led to significant improvements in automation, real-time data exchange, and decision-making. However, the rapid proliferation of connected devices also broadens the attack surface, making M2M systems vulnerable to sophisticated cyber threats. Traditional intrusion prevention techniques often struggle to adapt to the dynamic and distributed nature of these networks. In this study, we propose a reinforcement learning-based intrusion prevention framework tailored for M2M systems. By enabling autonomous agents to learn optimal defense strategies through continuous interaction with the environment, the system can proactively detect and mitigate malicious activities in real time. The model adapts to evolving threat patterns without relying on predefined signatures, making it particularly effective against zero-day attacks. Experimental results demonstrate that the proposed approach significantly improves detection accuracy, reduces false positives, and enhances the overall resilience of M2M communications. This work highlights the potential of intelligent, self-learning systems to secure future machine-driven ecosystems.

Keywords: Machine-to-Machine (M2M) Communication, Intrusion Prevention, Reinforcement Learning, Cybersecurity, Zero-Day Attacks, Anomaly Detection, Intelligent Systems

1. Introduction

1.1. Background and Motivation

Machine-to-Machine (M2M) communication has rapidly evolved into a foundational element of modern smart systems. By enabling direct data exchange between devices without human intervention, M2M underpins critical infrastructure such as autonomous vehicles, industrial control systems, and smart energy grids. These systems rely heavily on continuous, low-latency communication to ensure efficiency, automation, and responsiveness in dynamic environments.

As more physical systems become digitized, the integrity and security of the communication between machines grow increasingly vital. However, this evolution also introduces considerable security challenges. M2M networks often operate with limited resources, and many devices lack sufficient computational power to support robust cryptographic protection or real-time anomaly monitoring. Furthermore, M2M communications occur across highly heterogeneous platforms and unsecured networks, exposing the system to a wide range of potential cyber threats. Traditional cybersecurity tools such as signature-based intrusion prevention systems are no longer sufficient. They often fail to detect zero-day attacks, require frequent manual updates, and are typically reactive rather than adaptive. In contrast, intelligent security solutions—particularly those driven by machine learning—have shown promise in detecting sophisticated threats by identifying subtle patterns and anomalies. Among these, **Reinforcement Learning (RL)** stands out for its ability to adapt dynamically to evolving threats in real-time environments. RL agents learn optimal actions by trial and error, allowing them to recognize and respond to intrusions proactively rather than waiting for a predefined signature or pattern. Given the increasing attack surface in M2M systems and the limitations of static defense mechanisms, the use of reinforcement learning as an adaptive, autonomous approach to intrusion prevention has emerged as both a necessary and timely research direction.

1.2. *Research Problem*

Despite various advancements in M2M system design, a significant problem remains unsolved: **how to design an intrusion prevention system that is both adaptive and efficient enough to operate in real-time within constrained M2M environments**. Most existing security frameworks are either tailored to traditional computer networks or assume a level of processing capability and centralization that M2M systems typically lack. Furthermore, while many intrusion detection approaches focus on identifying threats after they have occurred, fewer efforts have been directed toward **proactive prevention**, especially in decentralized and dynamic environments. The few reinforcement learning models that do exist are generally designed for conventional IT networks and are often too resource-intensive or slow to deploy in edge-based or embedded M2M devices. There is, therefore, a pressing need to develop a lightweight, scalable, and context-aware intrusion prevention framework that can learn from its environment, adapt to

changing traffic behaviors, and make timely, accurate decisions to block malicious activity before it impacts system performance or safety.

1.3. *Objectives of the Study*

This study seeks to investigate and demonstrate how reinforcement learning techniques can be used to enhance intrusion prevention in M2M systems. The specific objectives include:

- To **model intrusion prevention as a reinforcement learning problem**, with clearly defined states, actions, and reward structures.
- To **design and implement a lightweight RL-based IPS** that can operate effectively within the constraints typical of M2M environments.
- To **evaluate the proposed model's accuracy, adaptability, and performance** relative to traditional intrusion prevention systems.
- To **analyze the impact of false positives and learning convergence** in high-variability network conditions.

1.4. *Contributions and Significance*

The key contributions of this study are as follows:

- A novel reinforcement learning framework is proposed that **transforms intrusion prevention into a real-time, sequential decision-making process**, allowing for dynamic policy updates as new threats emerge.
- The model is designed to be computationally efficient and **suitable for deployment in resource-constrained M2M environments** such as edge devices and gateways.
- A comparative analysis is presented, evaluating the **proposed model against traditional methods** (such as rule-based and supervised ML-based systems) in terms of detection accuracy, adaptability, and false positive rates.
- The research offers practical guidelines and implementation insights that bridge the gap between academic theory and real-world deployment in mission-critical M2M systems.

1.5. *Organization of the Paper*

The remainder of the paper is structured as follows:

- **Section 2** reviews the relevant literature, including an overview of M2M communication, its security challenges, and the role of RL in cybersecurity.
- **Section 3** outlines the proposed methodology, detailing the system architecture, RL model components, and evaluation metrics.
- **Section 4** presents the implementation results, including model training, performance analysis, and comparison with baseline systems.
- **Section 5** discusses the broader implications, strengths, limitations, and ethical concerns of the approach.
- **Section 6** concludes the study and offers directions for future research.

2. Literature Review

2.1. Overview of M2M Communication Systems

Machine-to-Machine (M2M) communication refers to the automated data exchange between devices, sensors, actuators, and control systems, typically without direct human intervention. M2M forms the backbone of many next-generation technologies, including industrial automation, smart cities, connected vehicles, and intelligent healthcare systems. These applications rely on the timely exchange of telemetry, control signals, and operational updates. M2M systems commonly use lightweight communication protocols such as MQTT, CoAP, or custom TCP/UDP variants to optimize for low power consumption and bandwidth. The network topologies in M2M environments may range from static to highly dynamic, and device roles may change at runtime based on contextual factors like mobility, failure recovery, or service demand. This decentralized, often autonomous nature allows for operational flexibility but makes consistent monitoring and control more difficult. In many deployments, particularly in edge-heavy or fog computing scenarios, devices must process, transmit, and react to data in real-time. Thus, ensuring data integrity and continuity of service is not just a matter of confidentiality—it is a functional and operational imperative. The increasing dependence on M2M for critical tasks such as remote diagnostics or autonomous navigation further highlights the importance of reliable and secure communication frameworks.

2.2. Security Challenges in M2M Environments

M2M networks face a host of security challenges due to their unique structural and operational characteristics. Unlike traditional enterprise systems, which may have dedicated security layers, M2M environments typically operate under resource constraints and in physically or logically distributed settings. This creates vulnerabilities on several fronts:

1. Device Heterogeneity and Constraints

M2M systems are composed of a diverse range of devices—some capable of advanced computation, others extremely limited. Implementing uniform security policies across this spectrum is inherently difficult. Many low-power sensors cannot support complex cryptographic algorithms or run real-time malware detection routines.

2. Insecure Communication Channels

Due to limited infrastructure or the use of proprietary or legacy protocols, many M2M communications occur over unsecured or semi-trusted networks. Attackers can exploit these channels through eavesdropping, packet injection, replay attacks, or man-in-the-middle (MITM) attacks.

3. Dynamic Topologies and Decentralization

In highly mobile or ad hoc deployments (e.g., vehicular M2M or mobile medical devices), network participants may frequently join or leave the system. This dynamicity renders static firewall rules and traditional network perimeters ineffective. It also increases the difficulty of maintaining updated blacklists, access control lists, or policy-based defenses.

4. Lack of Real-Time Adaptive Defenses

Conventional intrusion prevention systems (IPS) rely heavily on predefined rules and known attack signatures. While effective against familiar threats, they often fail to recognize novel or evolving attack patterns. Furthermore, these systems are typically designed for centralized infrastructures and may not scale or respond well in distributed M2M environments.

5. Physical Exposure and Tampering

Many M2M devices are deployed in unattended locations—remote weather stations, smart meters on utility poles, or public surveillance systems. This makes them vulnerable to physical

tampering, which can lead to malicious firmware injections or unauthorized access to the broader network.

In light of these challenges, the need for **autonomous, adaptive, and resource-aware security mechanisms** becomes apparent. Reinforcement learning, with its capacity for self-improvement and decision-making under uncertainty, provides a promising avenue for proactive intrusion prevention tailored to the specific needs of M2M environments.

3. Methodology

3.1. System Architecture and Assumptions

The proposed framework integrates a reinforcement learning (RL) agent into the control layer of an M2M network. The RL agent acts as a decision-making component within a lightweight Intrusion Prevention System (IPS) that monitors and evaluates incoming traffic in real-time.

Architecture Overview:

- **M2M Nodes:** Devices exchanging data over the network (sensors, actuators, gateways).
- **Traffic Monitor:** Captures network features such as packet headers, payload statistics, session duration, etc.
- **RL-Based IPS Agent:** Evaluates each connection or packet and determines whether it should be allowed or blocked.
- **Reward Engine:** Provides feedback to the RL agent based on the correctness of its action.

Assumptions:

- Devices may have limited resources, so model complexity must remain low.
- The environment is partially observable: not all features of the traffic or attack patterns are immediately visible.
- Attack patterns are not static and can evolve.

- A portion of labeled traffic data is available to bootstrap the RL agent (semi-supervised setting).

This architectural setup allows the RL agent to interact with the network in a continuous learning loop, improving its policy over time and adapting to new or previously unseen threats.

3.2. Reinforcement Learning Model Design

The core of the intrusion prevention mechanism is built on the principles of reinforcement learning, where the system learns to take the best action (e.g., allow or block traffic) based on the current state of network conditions and historical outcomes.

3.2.1. State Space and Action Space

State Space (S):

The state is defined as a vector of features extracted from incoming packets or network sessions. Example features include:

- Packet size
- Source/destination IP
- Protocol type (TCP, UDP, MQTT)
- Connection duration
- Frequency of packets from the same source
- Time since last communication from the source
- Number of failed authentication attempts

These features are normalized and encoded into a fixed-size vector for consistency and efficiency in computation.

Action Space (A):

The action space is discrete:

- **A = {0: Allow, 1: Block}**

  The simplicity of this action space ensures that the agent makes decisions quickly and decisively without unnecessary computational delay.

3.2.2. Reward Function Definition

The reward function is the key to shaping the behavior of the RL agent. It must incentivize accurate threat detection while penalizing false decisions.

The proposed reward structure:

- **+5** for correctly blocking a malicious packet (true positive)
- **+2** for correctly allowing a benign packet (true negative)
- **-3** for blocking a benign packet (false positive)
- **-5** for allowing a malicious packet (false negative)
- **-1** for indecision or delayed action

This asymmetric reward system reflects the real-world consequences of intrusion prevention. While both false positives and false negatives are undesirable, the system prioritizes minimizing false negatives due to their potentially severe impact.

3.2.3. Learning Algorithm (e.g., Q-Learning, DQN)

Two reinforcement learning algorithms were considered in this study:

1. **Q-Learning (Tabular):**

   Suitable for small, discrete state spaces. The Q-values are stored in a table and updated using the Bellman equation:

   $Q(s,a) \leftarrow Q(s,a) + \alpha [r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$ Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$Q(s,a) \leftarrow Q(s,a) + \alpha [r + \gamma a' \max Q(s',a') - Q(s,a)]$

   - α: Learning rate
   - γ: Discount factor

- r: Reward
- s, a: Current state and action
- s', a: Next state and best next action

2. **Deep Q-Network (DQN):**
   When the state space is too large or continuous for tabular methods, a DQN is used. It approximates the Q-function using a neural network with an input layer (for the state vector), hidden layers (ReLU activation), and output nodes (for each action's Q-value). The model is trained via stochastic gradient descent using the temporal difference error.

To stabilize training:

- **Experience replay** is used to store and sample past transitions.
- A **target network** is updated periodically to reduce instability during updates.

Both models were benchmarked, but DQN showed superior performance in more complex environments due to its generalization capability.

3.3. *Simulation Environment and Dataset*

To evaluate the proposed reinforcement learning-based intrusion prevention system, a controlled simulation environment was developed that emulates realistic M2M communication patterns and attack scenarios. The simulation platform was implemented using Python with support from **Scikit-learn**, **OpenAI Gym**, and **TensorFlow** for model training and evaluation.

Traffic Emulation:

The environment simulates communication between multiple M2M nodes, including sensors, edge controllers, and centralized gateways, exchanging telemetry and control messages. Each node generates normal traffic, such as periodic data uploads or actuator commands, and may occasionally encounter abnormal traffic due to simulated attacks.

Attack Injection:

Various intrusion types were synthesized into the environment:

- **Denial-of-Service (DoS):** High-frequency connection requests to exhaust bandwidth.
- **Spoofing:** Impersonation of legitimate nodes using falsified IP or MAC addresses.
- **Replay Attacks:** Reuse of previously captured legitimate packets to deceive the network.
- **Payload Manipulation:** Tampering with data fields in the communication payload.

Dataset Composition:

The model was trained and tested using a hybrid dataset composed of:

- **NSL-KDD dataset:** A well-known benchmark for intrusion detection research. While not originally built for M2M, relevant features were adapted to reflect M2M traffic behaviors.
- **TON_IoT telemetry dataset:** Provides realistic IoT/M2M traffic traces and attack samples across smart environments.
- **Synthetic M2M traffic:** Custom-generated flows mimicking lightweight protocols like MQTT, with embedded attack variants.

Each packet/session in the dataset is labeled as either benign or malicious, with features preprocessed through normalization and one-hot encoding where necessary. The dataset was divided into:

- 70% for training
- 15% for validation
- 15% for testing

This setup allowed the reinforcement learning model to experience a representative mixture of normal behavior and attack conditions, facilitating robust policy learning.

3.4. Performance Metrics

To objectively evaluate the performance and practical viability of the proposed RL-based intrusion prevention system, several metrics were used, reflecting both **security effectiveness** and **system efficiency**:

1. Detection Accuracy

The percentage of correctly identified traffic sessions (both benign and malicious) out of the total:

Accuracy=TP+TNTP+TN+FP+FN\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}Accuracy=TP+TN+FP+FNTP+TN

Where:

- **TP** = True Positive (attack correctly blocked)
- **TN** = True Negative (legitimate traffic allowed)
- **FP** = False Positive (legitimate traffic wrongly blocked)
- **FN** = False Negative (attack missed)

2. False Positive Rate (FPR)

An essential metric, especially in M2M systems where false alarms can disrupt critical machine functions:

FPR=FPFP+TN\text{FPR} = \frac{FP}{FP + TN}FPR=FP+TNFP

3. Convergence Time

The number of training episodes required for the RL model's performance (Q-values or policy) to stabilize. Faster convergence indicates the model can adapt to new environments or retrain efficiently.

4. Response Time

The average time taken by the model to make a decision (block/allow) after observing incoming traffic. Real-time systems demand sub-second response latency.

5. Resource Overhead (CPU/Memory)

Given the constrained nature of M2M nodes, the model's computational footprint was monitored. This ensures that the learning agent can run on modest hardware like embedded gateways or edge controllers.

6. Adaptability Score

A qualitative and quantitative metric measuring how well the model generalizes to novel or unseen attack types. This was assessed by introducing new attacks post-training and observing the change in accuracy without retraining.

Summary of Methodology

In summary, the methodology combined a thoughtfully designed RL agent, a representative hybrid dataset, and a simulation environment that mirrors real-world M2M communication. The chosen performance metrics allow for a well-rounded assessment of the system's reliability, responsiveness, and deployability in practical M2M scenarios.

4. Implementation and Results

4.1. Experimental Setup

The experimental environment was implemented using a virtualized testbed configured to replicate a distributed M2M network. Each simulated node represented typical IoT/M2M entities like sensors, actuators, and control units running lightweight communication protocols (e.g.,

MQTT, CoAP). The reinforcement learning agent was deployed at a virtual edge gateway responsible for monitoring and analyzing all incoming traffic.

Hardware configuration included:

- **Processor:** Intel Core i7 (8th Gen), 3.4GHz
- **Memory:** 16GB RAM
- **Software Stack:** Python 3.10, TensorFlow 2.x, OpenAI Gym, Scikit-learn, Wireshark for traffic capture

All simulations ran in a controlled Docker-based environment to maintain reproducibility. The RL agent used the **Double Deep Q-Network (DDQN)** for improved stability and faster convergence. Training was conducted over 5000 episodes, each representing a time window of M2M communication flows.

4.2. Model Training and Convergence

Initially, the model exhibited unstable decisions due to random action selection (exploration). As learning progressed, the agent began to distinguish attack patterns from benign flows. After approximately 1200 episodes, the Q-values stabilized, and the model converged to a reliable policy.

Key training observations:

- The ε-greedy policy was used with decaying exploration from ε=1.0 to ε=0.01.
- The loss function decreased steadily, indicating that the Q-network learned meaningful patterns.
- Rewards increased consistently, showing successful blocking of malicious activities.

Training convergence was reached after ~45 minutes of runtime. Post-convergence, the model was evaluated using the test dataset to validate its real-time prediction accuracy.

4.3. *Intrusion Detection Accuracy and Response Time*

The trained RL model demonstrated impressive classification capabilities in real-time. Against the test set, the following performance metrics were recorded:

Metric	Value
Accuracy	**96.2%**
Precision	95.5%
Recall	94.8%
F1-Score	95.1%
False Positive Rate	2.7%
Average Response Time	**87 ms**

The sub-100-ms response time confirmed that the agent could feasibly operate in real-time M2M systems, even on resource-limited devices. High F1-scores further signaled balanced performance across both detection and prevention.

4.4. *Comparison with Traditional Intrusion Prevention Systems*

To validate the superiority of the proposed approach, the RL agent was compared against two conventional intrusion prevention techniques:

- **Signature-based IDS (Snort)**
- **Rule-based anomaly detector (One-Class SVM)**

System	Accuracy	FPR	Response Time
Snort	89.7%	6.8%	250 ms
One-Class SVM	91.3%	5.1%	120 ms
Proposed RL Agent	**96.2%**	**2.7%**	**87 ms**

The reinforcement learning model consistently outperformed legacy solutions, especially in its lower false positive rate and faster decision-making. Traditional systems, while useful in static

environments, struggled to adapt to evolving attack vectors, highlighting the adaptability advantage of RL-based systems.

4.5. *Analysis of False Positives and Adaptability*

False positives were most common in scenarios involving bursty legitimate traffic, which occasionally mimicked DoS-like behavior. However, the model's false positive rate remained well below critical thresholds. To test adaptability, the model was exposed to new attack types such as model poisoning and encrypted payload injection, not present in the training set. Without retraining, the RL agent achieved a detection accuracy of **82.4%**, which demonstrates notable generalization capabilities and robustness.

5. Discussion

5.1. *Strengths and Limitations of the Proposed Framework*

Strengths:

- **Autonomous Adaptation:** The RL agent learns optimal defense strategies without manual rule-tuning.
- **High Detection Rate:** Outperforms traditional methods in accuracy and response speed.
- **Low False Positives:** Reduces system disruptions in mission-critical M2M applications.
- **Modular Deployment:** The agent can be integrated into existing M2M gateways or edge nodes.

Limitations:

- **Cold Start Problem:** Initial episodes suffer from poor detection until learning stabilizes.
- **Training Overhead:** Although online retraining is possible, it requires computational resources unsuitable for extremely constrained devices.
- **Scenario-Specific Tuning:** State and reward designs must be adjusted for different environments or communication protocols.

5.2. *Practical Implications for Real-World M2M Deployments*

The study demonstrates the viability of deploying intelligent intrusion prevention mechanisms in smart factories, vehicular networks, and healthcare monitoring systems where M2M communication dominates. By embedding the RL agent within edge nodes, intrusion responses become localized and real-time, minimizing latency and reducing reliance on central security servers. However, real-world deployment must consider network heterogeneity and data privacy concerns. Secure data sharing, model updates, and trust-based federated learning could help bridge these gaps.

5.3. *Scalability and Resource Constraints*

While the current implementation performs well on modest hardware, scaling to large M2M networks (thousands of nodes) introduces new challenges:

- **Policy Synchronization:** Distributed nodes may require a coordinated RL policy to avoid inconsistent behavior.
- **Energy Efficiency:** RL agents on battery-operated devices must optimize for power-aware computation.
- **Model Compression:** Techniques like pruning or quantization can reduce model size without sacrificing accuracy.

Edge-optimized RL models or hybrid federated learning frameworks may be more suitable in resource-constrained deployments.

5.4. *Ethical and Privacy Considerations*

Embedding intelligent intrusion prevention mechanisms in autonomous systems introduces ethical and privacy questions. Continuous monitoring of traffic, even for security purposes, may raise concerns over data collection practices, consent, and transparency.

Moreover, if misused or poorly configured, the RL agent could:

- Block legitimate services (availability risk)

- Learn unintended behaviors (bias in reward structure)
- Be exploited by adversaries to learn vulnerabilities (adversarial RL)

To mitigate these risks, clear privacy policies, model explainability, and regulatory compliance must be integrated into system design.

6. Conclusion and Future Work

6.1. *Conclusion*

This study proposed a reinforcement learning-based intrusion prevention framework tailored for machine-to-machine (M2M) communication systems. By integrating intelligent agents capable of learning from dynamic network behavior, the framework addresses the limitations of traditional security mechanisms that rely on static signatures or pre-defined rules. The proposed Double Deep Q-Network (DDQN) model demonstrated high detection accuracy (96.2%), fast response time (87ms), and low false positive rates, making it suitable for real-time M2M environments. Unlike conventional approaches, the RL agent adapts to emerging threats autonomously, optimizing prevention strategies as network conditions evolve. This makes it highly suitable for use in critical infrastructure such as smart grids, intelligent transportation systems, industrial IoT, and connected healthcare. However, challenges remain, particularly in minimizing computational overhead during training, handling the cold start problem, and maintaining privacy during traffic analysis. These considerations form the basis for ongoing refinement and future exploration.

6.2. *Future Work*

Several promising directions can enhance the effectiveness and practicality of this research:

- **Federated Reinforcement Learning:** Distributing learning across multiple edge devices without centralized data collection can improve scalability and privacy compliance.
- **Lightweight RL Models:** Adapting lightweight algorithms such as Q-Learning with function approximation or knowledge distillation techniques will enable deployment on ultra-low power devices.

- **Integration with Trust Models:** Incorporating trust evaluation into the RL agent's reward structure may offer better resilience against insider threats and deceptive node behavior.
- **Adversarial Robustness:** Future work should explore defenses against adversarial examples that may attempt to mislead the agent through carefully crafted traffic patterns.
- **Explainable RL:** Providing interpretable justifications for the agent's decisions is crucial for adoption in regulated environments. This includes visualizing state transitions and action-value trajectories.

The fusion of reinforcement learning with cybersecurity represents a transformative shift in how M2M systems can be safeguarded. As M2M continues to form the backbone of smart environments, building intelligent, self-adaptive defense mechanisms will remain a critical area of research and development.

References

1) Eziama, E., Ahmed, S., Ahmed, S., Awin, F., & Tepe, K. (2019, December). Detection of adversary nodes in machine-to-machine communication using machine learning based trust model. In *2019 IEEE international symposium on signal processing and information technology (ISSPIT)* (pp. 1-6). IEEE.
2) Kanthimathi, S. (2024). Exploring Machine learning algorithms for Malicious node detection using cluster based trust entropy. *IEEE Access*.
3) Ahmad, F., Kurugollu, F., Adnane, A., Hussain, R., & Hussain, F. (2020). MARINE: Man-in-the-middle attack resistant trust model in connected vehicles. *IEEE Internet of Things Journal*, *7*(4), 3310-3322.
4) Eziama, E., Jaimes, L. M., James, A., Nwizege, K. S., Balador, A., & Tepe, K. (2018, December). Machine learning-based recommendation trust model for machine-to-machine communication. In *2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)* (pp. 1-6). IEEE.
5) Eziama, E., Tepe, K., Balador, A., Nwizege, K. S., & Jaimes, L. M. (2018, December). Malicious node detection in vehicular ad-hoc networks using machine learning and deep learning. In *2018 IEEE Globecom Workshops (GC Wkshps)* (pp. 1-6). IEEE.

6) Volikatla, H., Thomas, J., Gondi, K., Indugu, V. V. R., & Bandaru, V. K. R. (2022). AI-driven data insights: Leveraging machine learning in SAP Cloud for predictive analytics. *International Journal of Digital Innovation*, *3*(1).
7) Volikatla, H., Thomas, J., Bandaru, V. K. R., Gondi, D. S., & Indugu, V. V. R. (2021). AI/ML-Powered Automation in SAP Cloud: Transforming Enterprise Resource Planning. *International Journal of Digital Innovation*, *2*(1).
8) Vummadi, J. R., & Hajarath, K. (2024). Integration of emerging technologies AI and ML into strategic supply chain planning processes to enhance decision-making and agility. *International Journal of Supply Chain Management*, *9*(2), 77-87.
9) Thomas, J., Vummadi, J., & Shah, R. (2024). Machine Learning Integrated Supplier Management Device. *Intellect. Prop. Off. is an Oper. name Pat. Off*.
10) Vummadi, J. R., & Krishna, C. R. H. (2024). Machine learning in SAP for inventory optimization. *International Journal of Supply Chain Management: Vol. Vol*, *9*.
11) Abbasi, A., & Yampolskiy, M. (2021). *Machine learning for intrusion detection in industrial control systems: A survey*. Computers & Security, 105, 102240. https://doi.org/10.1016/j.cose.2021.102240
12) Alrawais, A., Alhothaily, A., Hu, C., & Cheng, X. (2017). *Fog computing for the Internet of Things: Security and privacy issues*. IEEE Internet Computing, 21(2), 34–42. https://doi.org/10.1109/MIC.2017.37
13) Bedi, H. S., & Bajaj, K. (2020). *Reinforcement learning-based anomaly detection for IoT communication networks*. Journal of Intelligent & Fuzzy Systems, 38(2), 1911–1920. https://doi.org/10.3233/JIFS-179103
14) Fu, Y., Yu, F. R., Liu, J., Luan, T. H., & Zhang, Y. (2018). *Intrusion detection for intelligent transportation systems using reinforcement learning*. IEEE Wireless Communications, 25(6), 112–118. https://doi.org/10.1109/MWC.2018.1700324
15) Kwon, D., & Kim, H. (2020). *A deep reinforcement learning-based defense framework for autonomous vehicular networks*. Sensors, 20(11), 3206. https://doi.org/10.3390/s20113206
16) Liu, H., Lang, B., Liu, M., & Yan, H. (2020). *CNN and RNN based payload classification methods for attack detection*. Knowledge-Based Systems, 163, 332–341. https://doi.org/10.1016/j.knosys.2018.09.011

17) Moosavi, S. R., Gia, T. N., Nigussie, E., Rahmani, A. M., Virtanen, S., & Tenhunen, H. (2017). *End-to-end security scheme for mobility-enabled healthcare Internet of Things*. Future Generation Computer Systems, 64, 108–124. https://doi.org/10.1016/j.future.2016.02.005

18) Sagar, S., & Dey, N. (2021). *Securing M2M communication using deep learning models: A review*. In Deep Learning for Internet of Things Infrastructure (pp. 115–134). Springer. https://doi.org/10.1007/978-981-15-7743-1_6

19) Sultana, T., Chilamkurti, N., & Peng, W. (2019). *Survey on SDN based network intrusion detection system using machine learning approaches*. Peer-to-Peer Networking and Applications, 12(2), 493–501. https://doi.org/10.1007/s12083-017-0630-0

20) Xu, K., Shen, H., & Wang, H. (2020). *A novel reinforcement learning approach for anomaly detection in networked systems*. IEEE Transactions on Network Science and Engineering, 7(4), 2667–2677. https://doi.org/10.1109/TNSE.2019.293345