International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2025/12 Volume 12, Issue 2, March – April (2025)

RESEARCH ARTICLE

Design of an Augmented Cyber Attack Detection Model for Securing IoT Kernels Via Deep Dyna Q and VARMA GRU-Based Predictive Analysis

Bharat S. Dhak
Computer Science and Engineering Department, HVPM's College of Engineering and Technology, Amravati, India.
✉ bharat.dhak@gmail.com

Prabhakar L. Ramteke
Computer Science and Engineering Department, HVPM's College of Engineering and Technology, Amravati, India.
pl_ramteke@rediffmail.com

Received: 11 December 2024 / Revised: 13 February 2025 / Accepted: 18 March 2025 / Published: 30 April 2025

Abstract – The Internet of Things has made our lives easier to live and more convenient. However, the risks of cyber-attacks, especially within the kernels of the IoT, have increased manifold. A strong security system is required to ensure that the devices are safe from any threat. Therefore, proposing an augmented pattern evaluation model that makes use of Deep Dyna Q and VARMA GRU-based predictive analysis to be able to provide additional embedded security features to IoT kernels. Altogether, model that composes of three components: feature extraction, model training, and prediction operations. At the first component, extraction of the relevant features from the IoT kernel in order to create a feature vector for this process is carried out. The feature vector is further utilized to train the Deep Dyna Q algorithm, a reinforcement learning approach which learns to decide under the maximization of some reward signal. Here the use the second module with the VARMA GRU-based predictive analysis algorithm to predict the future state of the IoT kernel based on the current state and actions taken by the Deep Dyna Q algorithm. The VARMA GRU algorithm implements a VARMA with the advantages of a GRU model and thus provides forecasts accurately. In the final component, assess the predicted state of the IoT kernel with a set of predefined security rules. If any of these rules are broken by the predicted state, the system acts accordingly to reduce the possible threats. This will be the model that would consider an all-encompassing security of IoT kernels by harnessing the power of various algorithms. The Deep Dyna Q ensures that the system will make intelligent decisions in real-time, while the VARMA GRU adds accuracy with its predictive analysis algorithm, hence making this an augmented pattern evaluation model that would rise to the ever-increasing security challenges of IoT devices and deployments.

Index Terms – IoT, Security, Kernel, Complexity, Delay, Scalability, MRM, QoS, Performance, Machine Learning, Blockchain, SIDECHAIN, Encryption, Hashing, Key, Communications.

1. INTRODUCTION

With the unabated popularity of the Internet of Things within the past few years, IoT actually served to connect devices and automate them in such a way as to make our lives easier. However, this widespread use of IoT devices also brought to us associated security concerns connected with the security in kernels of IoT devices. Basically, IoT kernels [1, 2, 3] are considered the heart of any IoT device and control and manage all other components. In this respect, the security of IoT kernels forms the basis for securing the entire IoT devices and deployments. The security systems developed for IoT kernels still mostly depend on a signature-based approach, which is helpful in detecting known threats but falls short in detecting unknown or zero-day attacks. This therefore means there is a need to develop a more comprehensive security system that can detect and prevent such attacks. Machine Learning and Predictive Analytics have thus emerged in recent years as a formidable solution to the challenges traditional Signature-Based approaches have faced. In this context [4, 5, 6], the authors suggest an augmented pattern evaluation model that integrates the strength of the Deep Dyna Q with the VARMA-GRU-based predictive analysis algorithms to improve security for IoT kernels. The Deep Dyna Q algorithm is a kind of reinforcement learning that learns to decide by optimizing an accumulated reward signal. It is very suitable in real-time decision scenarios due to its ability to adapt itself in changing environments. On the other side, the VARMA GRU-based predictive analysis algorithm synergizes the strengths of both VARMA and GRU models in order to provide an accurate prediction of the future states of the IoT kernels. A predictive model is proposed, which includes three main parts: feature extraction, model training, and prediction. First of all, this is extracting relevant features

RESEARCH ARTICLE

from the IoT kernel to create a feature vector. The Deep Dyna Q algorithm learns to decide under this very reward signal, with the feature vector fed into it. Given the current state and the actions, the Deep Dyna Q algorithm will take, the second component of the VARMA GRU-based predictive analysis algorithm will predict the future state of the IoT kernel. Finally, the predicted state of the IoT kernel is checked against predefined security rules to detect threats. Therefore, this paper proposes an augmented pattern evaluation model that empowers the potential of machine learning and predictive analytics in order to enhance the security features of IoT kernels. It can help alleviate the growing security challenges to IoT devices with the proposed model and detect unknown or zero-day attacks. The rest of the paper is organized as follows: Section 2 reviews related work, Section 3 describes the proposed model in detail, Section 4 presents the experimental results, and finally, Section 5 concludes the paper with a summary of the contributions and future recommendations.

The objectives of the paper are,

1. To identify the challenges in securing IoT kernels and to highlight the limitations of current security systems.
2. Propose an augmented pattern evaluation model that will further leverage the power of machine learning and predictive analytics in enhancing security for IoT kernels.
3. Explain various algorithms that constitute this model, notably Deep Dyna Q and VARMA GRU-based predictive analysis, and describe how they work together to strengthen security in an IoT kernel.
4. Presentation of results of the experiments carried out on proposed model and comparison of the performance with already existing security systems in current scenario.
5. Proof of effectiveness of proposed model in detection and prevention of unknown or zero-day attacks on IoT kernels.
6. Contribute to this line of work in developing a more comprehensive security system for IoT devices and provide insights for future works in this area of real-time deployment.

The recent growth of use of IoT has led to vulnerability in terms of security especially regarding kernel-level in which most core operations of IoT devices are kept. Traditional security mechanisms, mainly relying on signature-based intrusion detection, fail to identify novel or zero-day attacks. Thus, the IoT deployments become vulnerable to adversarial threats. Moreover, the existing security models have high computational complexity, resource-intensive operations, and lack real-time adaptability, which makes them inefficient for constrained IoT environments. Thus, there is a serious requirement for an advanced, intelligent security model that can be proactive in threat detection and mitigation with low computation overheads. Herein, the paper proposes an augmented pattern evaluation model that combines Deep Dyna Q reinforcement learning with VARMA-GRU-based predictive analytics to ensure real-time and accurate threat assessment and response within IoT kernels.

The rest of the paper will follow the following outline: In Section 2, a comprehensive literature review on relevant works in the area of IoT kernel security is provided that reviews the existing prevalent challenges and shortfalls in existing literature. Finally, Section 3 will outline the designed architecture and approach of the proposed augmented pattern evaluation model with its integration of Deep Dyna Q and VARMA-GRU-based predictive analysis. Section 4 outlines the experimental setup, dataset selection, evaluation metrics, and comparative analysis of the approach proposed with those existing methods. Section 5 discusses the outcome, highlighting improvement in performance with reduced false alarms and enhanced mitigation capabilities. To end, the paper is finalized with Section 6, where overall contributions and areas for further expanding IoT security will be provided and possible future works.

2. RELATED WORK

This literature review on the current state of IoT security encompasses information on the existing landscape, challenges, and new developments around the security of IoT environments. To that end, each study has its own methodologies, findings, and limitations; hence, it offers very valuable insights into the dynamically changing field of IoT security. Convergence of different machine learning techniques, advanced encryption methods, and hybrid frameworks underlines a multi-dimensional approach to dealing with complex security issues inherent in IoT systems. Geetha et al. [1] propose an adaptive weighted kernel support vector machine-based circle search approach that ensures a very high detection accuracy of about 95%. Nevertheless, as shown in table 1, the high computational cost throws a spanner into the works for real applications with resource-constrained devices. Likewise, Quincozes et al. [2] present a survey of IoT protocols at the application layer and discuss some of the already existing security-related issues and how explainable AI can be exploited in such scenarios. Although these findings are insightful, a major gap exists in terms of the lack of implementation. Li and Dou's [3] active eavesdropping detection increased the detection rate by 20%, although it only worked on the physical layer of security; therefore, complementary measures are needed.

In improving IoT security, Kumar et al. [4] add a security protocol in a Zigbee network-specific way through their analysis on IoT security with Zigbee network-based security; therefore, broader application is needed. Ul Haq et al. [5] produce a finer review on IoT firmware security but without experimental validation, weakening its impact. Rana et al. [6]

RESEARCH ARTICLE

adopted krill-based deep neural network stacked autoencoders for cyber threat detection, hence achieving an accuracy of 93%, though at high training complexity. Om Kumar et al. [7] used recurrent kernel convolutional neural networks to improve intrusion detection, thus attaining 91% accuracy, but at very high resource usages. Mohy-eddine et al. [8] deal with artificial neural networks for smart farming security and obtain a detection rate of 89%, though with little generalizability outside smart farming. Kaliappan et al. [9] proposed an AI-based trust framework that is capable of enabling intrusion detection and improving the accuracy to 92%.

Karamizadeh et. al [10] proposed combination of deep learning and self- attention mechanism which enhanced smart home security. Rajarajan et. al. [11] work shows hybrid optimized learning model for intrusion classification which enhanced classification accuracy and achieved 93% detection accuracy rate. In the work of Zada et al., [12] kernel ELM and war strategy optimization have been adopted to accomplish software defect prediction with 87% accuracy; however, the work is limited to software defects. Boopathi et al. [13] have tried optimization algorithms for privacy-preserving data disturbance in edge computing, whereby privacy and security are improved but at a high complexity. The fitness tracker security using quantum fruit fly optimization by Shanthala and Annapurna [14] has an accuracy of 90%, specific to the fitness trackers. Hazman et al. [15] use ensemble learning for smart environment security, improving intrusion detection with a rate of 88% but at high resource requirements. The transfer learning model by Nandanwar and Katarya [16] enables intrusion detection in IoT with a precision of 91%, though at the cost of extensive training data. While Zhan et al. [17] focus on fine-grained kernel access limitation, this greatly reduces the attack surface but at quite a cost in implementation complexity. In the case of third-party services, Jang and Kang's [18] trusted execution environment for IoT services can improve security, although at a rather high cost to implement. Tang et al.'s [19] jammer- assisted secure precoding scheme for MIMO IoT networks enhances physical layer security and improves the secrecy rate but only applies for MIMO networks. Zhu and Tang [20] proposed an NB-IoT-based remote SSH access that can ensure secure access to UAVs, but it is specific to UAV applications. Oliveira et al. [21] propose an open-source TEE for IoT devices aiming to improve security and isolation, enhancing the security architecture, while putting forward several implementation challenges. Bedari et al.[22] have put forward a feature transformation-based fingerprint authentication system that effectively improves privacy protection and the accuracy of authentication but increases computational costs. Feng et al. [23] give a comprehensive review of IoT firmware vulnerability detection, identifying some of the key vulnerabilities without experimental results in their study. Takemura et al. [24] enhance auditability and system reliability of edge device provenance auditing with high TEE implementation complexity. Li et al. [25] enhance the speed of encryption performed on IoT systems via GPU-accelerated homomorphic encryption at a high hardware resource cost. Wu et al. [26] enhance security in RF environments via RF fingerprint recognition in low-SNR settings, which ensures high accuracy in detection but is confined to RF scenarios. Hwang et al.'s [27] runtime framework for trusted applications in ARM/FPGA systems features high security and high performance but is very complicated. Li and Takada's [28] hypervisor middleware for IoT systems ensures high reliability and security, while implementation remains troublesome. High-grade security for updates and decent firmware management are provided at high resource costs by the micro-kernel approach of Aspesi and Zaccaria [29] in update management firmware.

Sheybani et al. [30] provide secure hashing for sensor values, which enhances security at the source with a high computational overhead to ensure data integrity. Ning et al. [31] describe the defense mechanisms against debugging features of the ARM architecture, thereby enhancing the security and reducing the risk of privilege escalation. however, this work is restricted to ARM architecture. Kaiser et al. [32] present an exhaustive review of the container technologies available for the ARM architecture, outlining major security features without detailing practical implementation operations. Predictive models for personalized diabetes monitoring system proposed by R. Marzouk et. al. [33] has improved health monitoring as well as predictive accuracy but which is limited to diabetes monitoring. Park et al. [34] presented an investigation into security control for inference systems, improving intrusion detection and the security of a system at high complexity. Li et al. [35] provided defense against poisoning attacks in federated learning, which enhances model robustness at large resource usages. He et al. [36] further adopted federated learning for edge device identification to improve identification accuracy and network security but at high computational costs.

Iqbal et al. [37] contribute to ransomware detection in IoT healthcare systems; the security is enhanced but only in relation to healthcare. Gyamfi and Jurcut [38] perform intrusion detection in industrial IoT for the enhancement of the accuracy of detection and system security, but it has high implementation complexity. Wang et al. [39] put forward distributed classification learning against attacks involving flipping labels to improve the robustness of learning and classification accuracy at a high computational cost. Alruwaili et al. [40] applied probabilistic transfer learning in the monitoring of data transmission from wearable sensors, which enhances the accuracy of monitoring and the security of transmission but limits it to wearable sensors.

RESEARCH ARTICLE

Table 1 Empirical Review of Existing Methods

Reference	Method Used	Findings	Results	Limitations
[1]	Adaptive weighted kernel support vector machine-based circle search approach	Enhanced intrusion detection accuracy	Achieved 95% detection accuracy	High computational cost
[2]	Survey on IoT application layer protocols and security challenges	Identified key security challenges and the role of explainable AI	Provided comprehensive analysis	Lack of practical implementation
[3]	Active eavesdropping detection at the physical layer	Improved detection of eavesdropping attacks	Increased detection rate by 20%	Limited to physical layer security
[4]	Comprehensive analysis for Zigbee network-based IoT security	Identified vulnerabilities in Zigbee networks	Improved security protocols	Specific to Zigbee networks
[5]	Survey on IoT firmware security	Reviewed extraction techniques and vulnerability analysis frameworks	Comprehensive survey results	No experimental validation
[6]	Krill-based deep neural network stacked auto encoders	Enhanced detection of cyber threats	Achieved 93% accuracy	High training complexity
[7]	Recurrent kernel convolutional neural network for intrusion detection	Improved model accuracy and speed	Achieved 91% detection accuracy	High resource usage
[8]	Artificial neural network for smart farming security	Enhanced detection of malicious activities in farming	Achieved 89% detection rate	Limited generalizability
[9]	AI-based trust framework for intrusion detection	Improved trust and security in IoT systems	Achieved 92% accuracy	High implementation complexity
[10]	Combination of deep learning and self-attention mechanism	Enhanced smart home security	Achieved 94% accuracy	High computational requirements
[11]	Hybrid optimized learning model for intrusion classification	Enhanced classification accuracy	Achieved 93% detection accuracy	High computational overhead
[12]	Kernel ELM and war strategy optimization for software defect prediction	Improved defect prediction accuracy	Achieved 87% accuracy	Limited to software defects
[13]	Optimization algorithms for privacy-preserving data disturbance	Enhanced privacy and security in edge computing	Achieved significant privacy gains	High complexity
[14]	Quantum fruit fly optimization for fitness tracker security	Enhanced security for fitness trackers	Achieved 90% accuracy	Specific to fitness trackers
[15]	Ensemble learning for smart environment security	Improved intrusion detection	Achieved 88% detection rate	High resource requirements
[16]	Transfer learning model for intrusion detection in IoT	Enhanced prediction accuracy	Achieved 91% accuracy	High training data requirements

RESEARCH ARTICLE

[17]	Fine-grained kernel access limitation	Enhanced OS security for IoT systems	Reduced attack surface significantly	Implementation complexity
[18]	Trusted execution environment for IoT services	Enhanced security for third-party services	Improved service security	High implementation cost
[19]	Jammer-assisted secure precoding for MIMO IoT networks	Improved physical layer security	Enhanced secrecy rate	Specific to MIMO networks
[20]	NB-IoT-based remote SSH access	Improved secure access to UAVs	Enhanced security	Limited to UAVs
[21]	Open-source TEE for IoT devices	Enhanced security and isolation	Improved security architecture	Implementation challenges
[22]	Feature transformation-based fingerprint authentication	Enhanced privacy protection	Improved authentication accuracy	High computational cost
[23]	Survey on vulnerability detection in IoT firmware	Comprehensive review of firmware security	Identified key vulnerabilities	No experimental results
[24]	TEE for provenance auditing on edge devices	Enhanced auditability and security	Improved system reliability	High implementation complexity
[25]	GPU-accelerated homomorphic encryption	Enhanced encryption for IoT systems	Improved encryption speed	High hardware requirements
[26]	RF fingerprint recognition in low-SNR settings	Improved security through RF recognition	Enhanced detection accuracy	Limited to RF environments
[27]	Runtime framework for trusted applications in ARM/FPGA systems	Enhanced security for hybrid systems	Improved security and performance	High complexity
[28]	Hypervisor middleware for reliable IoT systems	Improved system reliability	Enhanced middleware security	Implementation complexity
[29]	Micro-kernel approach for firmware updates	Enhanced update security	Improved firmware management	High resource requirements
[30]	Secure hashing for sensor values	Enhanced security at the source	Improved data integrity	High computational overhead

This comprehensive review is a showcase for IoT security with its remarkable improvements through a variety of techniques or methodologies and their results. While each contributes to the general understanding of effective IoT environment protection, many have limitations that models proposed here address. The integration of Deep Dyna Q into VARMA GRU-based predictive analysis makes this model robust and efficient for IoT security, which is further evidenced by its performance in almost all metrics evaluated against the methods reviewed. Along this line of argument, the proposed model, with an overall accuracy of 97.8%, outperforms the highest reported accuracy of 95% by Geetha et al., hence better at intrusion detection. The model also reaches an FPR of 1.85% and an FNR of 1.35%, considerably improved over that of other methods in terms of FPR and FNR. The average response timestamp for the proposed model was also way faster, 120 ms, in comparison with the response times of 300 ms, 250 ms, and 350 ms observed for the other methods. Fast response capability is necessary to counter these threats before the exploitation of vulnerabilities across a variety of high-consequence scenarios.

It is also indicated by the resource utilization metrics that the model is quite efficient: 25% CPU, 150 MB of memory, additional network overhead of 12 KB/s—each well below comparative methods. The high mitigation success rate of 95% with an average mitigation timestamp of 180 ms and a very low user impact (on a scale of 1-10) of 2 presents a spotlight for the robust and user-friendly approach of the model toward threat mitigation. These results thus validate the

RESEARCH ARTICLE

efficacy and efficiency of the proposed model for enhancing the security at the kernel level in IoT devices. In the future, more sophisticated machine learning algorithms and ensemble methods can be integrated to further reduce false positive and negative rates. Increasing the dataset size by including further diversified and complex IoT scenarios will improve its generalizability and robustness. Other significant developments in the area are real-time adaptive learning methodologies, blockchain technologies for securely logging events, and context-aware analysis. This paper is, therefore, also a timely move to ensure that the changing compliance of both regulatory as well as privacy standards would become practical. There is also the balance between security and usability, which comes through user-centered studies in order for implementations to get the right acceptance. Conclusion: The imperative need for stringent IoT security safeguards cannot be overemphasized, which is one important aspect as further supported according to the overview of the relevant recent literature submitted above. Although a lot has been done on this topic, the model proposed here offers a solution of high efficiency against the continuous challenges in IoT security. This approach ensures improved efficiency in terms of detection accuracy, response times, and usage of resources while being user-friendly. Such innovative approaches will have a place as the IoT landscape continues to further evolve in terms of preserving the integrity and security of interconnected systems.

3. PROPOSED SYSTEM WORKFLOW

3.1. Design of an augmented Cyber Attack Detection Model for securing IoT Kernels via Deep Dyna Q & VARMA GRU-based Predictive analysis

This is a three-phase model for the IoT Kernal. The workflow is based on feature extraction, decision-making with reinforcement learning, and predictive analytics.

3.1.1. Workflow: IoT Kernel Security and Threat Detection

Phase 1: Feature Extraction

Objective: Extract and pre-process relevant features from the IoT system.

1. Input: Raw data from the IoT system (CPU usage, network traffic, sensor readings).
2. Extract Features:
- Collect and measure key features such as CPU usage, network traffic, and sensor data.
- Ensure the extracted features reflect critical state information of the IoT system.
3. Pre-process Features:
- Normalize the extracted features (e.g., scaling, handling missing values).
- Filter out noise and irrelevant data to keep only the necessary features.
4. Create Feature Vector:
- Combine the pre-processed features into a structured feature vector.
- The feature vector should represent the current state of the IoT kernel.
5. Feed to Model:
- Feed the structured feature vector as input into the next phases of the model.

Phase 2: Decision-Making with Deep Dyna Q (Reinforcement Learning)

Objective: Use reinforcement learning to make intelligent security decisions.

1. Input: Feature vector from Phase 1, current state of the system.
2. Initialize Model:
- Use the Deep Dyna Q algorithm, which combines real and simulated experiences.
- Initialize Q-values and policy for security decision-making.
3. Decision Process:
- Evaluate the current system state using the feature vector and make a decision (e.g., allow or block network traffic, or trigger an alert).
- Apply a reward function that:
  - Rewards: States indicating secure and normal operations.
  - Penalizes: States with abnormal activity or potential security threats.
4. Update Policy:
- Continuously update the decision-making policy using the real-time feedback from the environment (i.e., system's response to actions taken).
- Use simulated experiences to augment learning in addition to real-world experiences.
5. Iterative Learning:
- Repeat the process iteratively to refine the policy over time for optimal security management.

RESEARCH ARTICLE

- The model should balance between mitigating security threats and ensuring system performance is not overly degraded.

Phase 3: Predictive Analytics with VARMA-GRU Models

Objective: Predict future states of the IoT kernel and trigger actions if a threat is detected.

1. Input: Time-series data (from Phase 1 and Phase 2 outputs).
2. Build VARMA-GRU Model:

- Use the VARMA (Vector Auto Regressive Moving Average) model to capture linear dependencies in the time-series data.
- Use the GRU (Gated Recurrent Unit) model to capture non-linear trends and adapt to system dynamics.

3. Predict Future States:

- Use the combined VARMA-GRU model to predict the future state of the IoT kernel (e.g., CPU usage, network traffic, or sensor behaviour).

4. Compare to Security Rules:

- Compare the predicted future state to predefined security rules or thresholds.
- If any of the predicted states violate these security rules (e.g., indicate a possible threat or anomaly), trigger an alert.

5. Mitigation:

- Apply appropriate mitigation actions based on the triggered rules (e.g., isolate suspicious devices, block certain network traffic, or trigger a security protocol).

The very first phase of this algorithm involves feature extraction from CPU usage up to network traffic and sensor readings. The extracted features are therefore preprocessed, structured into a feature vector, and used to feed the model with learning inputs. The extracted features are very crucial so that the model covers all critical state information while removing only noise or unwanted data. The Deep Dyna Q reinforcement learning algorithm is used for making intelligent security decisions in the second phase. Deep Dyna Q, as opposed to traditional Q-learning, uses real as well as simulated experiences to optimize dynamic decision-making. The model includes a reward function that encourages secure states and penalizes any anomalous activities that might be indicative of a security threat. The reinforcement model improves its policy in real time through iterative learning processes to detect and mitigate threats without causing unwarranted interventions that could degrade system performance. Predictive analytics utilizes VARMA-GRU models to predict the future state of the IoT kernel in the final phase. The VARMA model captures linear dependencies in time-series data, whereas the GRU network captures the non-linear trend, which would provide a proper adaptive threatening prediction. If such a predicted state violates predefined security rules, mitigation actions are applied by the appropriate rule engine for such a triggering. This architecture significantly improves threat detection accuracy, and the pro-active defense mechanism can work to counter emerging threats in the setup of IoT. This section describes a design for an augmented pattern evaluation model to secure IoT kernels using deep Dyna Q and VARMA GRU-based predictive analysis, which would allow the overcoming of the low efficiency and high complexity issues in existing models. First, according to figure 1, in the context of IoT kernel security, this very important step in creating a good feature vector is feature extraction. This will lay down the basis for training the Deep Dyna Q algorithm from IoT kernel data. The process of feature extraction involves the analysis of data from the IoT kernel for critical attributes which capture the state and behavior of the system underlying it. In this process, statistical techniques and domain knowledge are used to ensure that the feature vector is complete and representative. Let X(t) be the state vector of the IoT kernel at timestamp t sets. Extracting features from it starts with gathering raw data D(t) from the kernel, which contains parameters such as the amount of CPU in use, memory consumption, network traffic, and sensor readings. Equation (1) processes the raw data to remove noise and normalize the values, returning a cleaned dataset, D′(t),

$$D'(t) = Normalize\left(RemoveNoise\left(D(t)\right)\right) \quad (1)$$

The computation of statistical features from cleaned dataset samples follows. Let these be the mean, μ variance, σ^2, skewness, γ1, and kurtosis, γ2, of time-series data samples. Statistical moments capture succinctly an excellent summary of the distribution and the shape of data, capturing essential patterns via equations (2, 3, 4, & 5):

$$\mu i = \frac{1}{T}\sum_{t=1}^{T} Di'(t) \quad (2)$$

$$\sigma i^2 = \frac{1}{T-1}\sum_{t=1}^{T} (Di'(t) - \mu i)^2 \quad (3)$$

$$\gamma(1,i) = \frac{1}{T}\sum_{t=1}^{T} \frac{(Di'(t) - \mu i)^3}{\sigma i^3} \quad (4)$$

$$\gamma(2,i) = \frac{1}{T}\sum_{t=1}^{T} \frac{(Di'(t) - \mu i)^4}{\sigma i^4} - 3 \quad (5)$$

RESEARCH ARTICLE

The statistical features computed above form the preliminary feature vector F(t), which can be further enhanced by incorporating domain knowledge samples. This may include certain patterns of security breaches, for example, sudden spikes in network traffic or some other suspicious changes in sensor readings. Further features, Fd(t), capturing such security-relevant patterns can be introduced by domain experts using equation (6).

$$F(t) = [\mu i, \sigma i^2, \gamma 1, i, \gamma 2, i, Fd(t)] \quad (6)$$

This feature vector, F(t), trains the Deep Dyna Q algorithm process. Deep Dyna Q is a reinforcement learning approach that combines Q-learning with a model-based planning component. The objective of this algorithm is to maximize a reward signal R(t), designed to reflect the security state of the IoT kernel. The Q-learning update rule is given via equation (7),

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left(R(t) + \gamma \max_{a'} Q(s',a') - Q(s,a)\right) \quad (7)$$

Where s is the current state, a is the action taken, α the learning rate, γ the discount factor, and s′ the next state for this process. In the definition of the reward signal R(t), it is according to the security evaluation of the kernel state, giving higher rewards to those states that are considered secure for this process. Deep Dyna-Q's model-based component involves learning a transition model, T'(s,a), and reward model, R'(s,a), predicting the next state and reward respectively. These models are trained on the basis of feature vector F(t) and observed transitions, and rewards. The transition model is updated using supervised learning techniques via equations (8 & 9).

$$T'(s,a) \leftarrow UpdateModel(T'(s,a), F(t), s') \quad (8)$$

$$R'(s,a) \leftarrow UpdateModel\big(R'(s,a), F(t), R(t)\big) \quad (9)$$

These models are used by the Deep Dyna Q algorithm in simulating future trajectories for the refinement of its policy and therefore the decision-making capabilities regarding securing an IoT kernel. The approach was chosen for its ability to conflate model-free and model-based reinforcement learning into a trade-off between exploration and exploitation. Deep Dyna Q enables real-time intelligent decisions of a system while continuing to improve upon its performance through the use of simulated experiences. These feed very well into the VARMA-GRU-based predictive analysis that provides an accurate forecast of the state of the IoT kernel to enable proactive security measures. The above methods—pairing this solution—offer a robust solution to dynamic and complex security challenges of IoT devices and deployments.

Basically, the algorithm for predictive analysis using VARMA GRU foretells the future status from an IoT kernel device concerning the current state and actions passed through the Deep Dyna Q algorithm. To be specific, this predictive model has integrated strengths from a vector autoregressive moving average model with those of the gated recurrent unit in modeling linear dependencies and complex temporal dynamics on data samples. It begins by employing the VARMA model in modeling the linear relationships between the multivariate time-series data samples. Let X(t) be the state vector of the IoT kernel at timestamp, t sets. The VARMA model is defined via equation (10).

$$X(t) = \sum_{i=1}^{p} AiX(t-i) + \sum_{j=1}^{q} Bj\epsilon(t-j) + \epsilon(t) \quad (10)$$

Where Ai and Bj are coefficient matrices, and p and q are the orders of the autoregressive and moving average components respectively; is a white noise vector for this process. The VARMA model captures linear dependencies in state vector X(t), which can be used to provide a baseline prediction. Paper represent the residuals from the VARMA model via equation (11).

$$R(t) = X(t) - X'VARMA(t) \quad (11)$$

These residuals are fed into the GRU network. The GRU is a type of recurrent neural network that may model challenging time patterns and capture long-term dependencies in data samples very well. It has two gates: a reset gate rt and an update gate zt, working together to mediate the flow of information across the network. The reset gate is defined via equation (12),

$$rt = \sigma(WrR(t) + Urh(t-1) + br) \quad (12)$$

Where, Wr and Ur are weight matrices, h(t−1) is the hidden state from the previous time stamp, and br is a bias vector for this process. Equation (13) defines the update gate as,

$$zt = \sigma(WzR(t) + Uzh(t-1) + bz) \quad (13)$$

The candidate hidden state h~t is then computed via equation (14).

$$h{\sim}t = tan\,h(WhR(t) + rt \odot Uhh(t-1) + bh) \quad (14)$$

Where, ⊙ represents the element-wise multiplication for this process. The final hidden state ht is obtained by combining the previous hidden state and the candidate hidden state, modulated by the update gate via equation (15).

$$ht = (1 - zt) \odot h(t-1) + zt \odot h{\sim}t \quad (15)$$

The output of the GRU network, R'(t), represents the non-linear residual prediction, which is then combined with the VARMA prediction to obtain the final forecast via equation (16).

RESEARCH ARTICLE

$$X'(t) = X'VARMA(t) + R'(t) \quad (16)$$

VARMA-GRU was selected for the predictive analysis algorithm since it is capable of picking up both linear and nonlinear dependencies in samples of IoT kernel state data. The linear elements will be treated by the VARMA model, while complicated patterns of time series trends and nonlinearities will be captured by the GRU network, which is a complementary combination to ensure the robustness and accuracy of the prediction for the future state of the IoT kernel. As explained in figure 2, the last part of the proposed model is the use of a rule engine to check the predicted state of the IoT kernel against a set of predefined security rules.

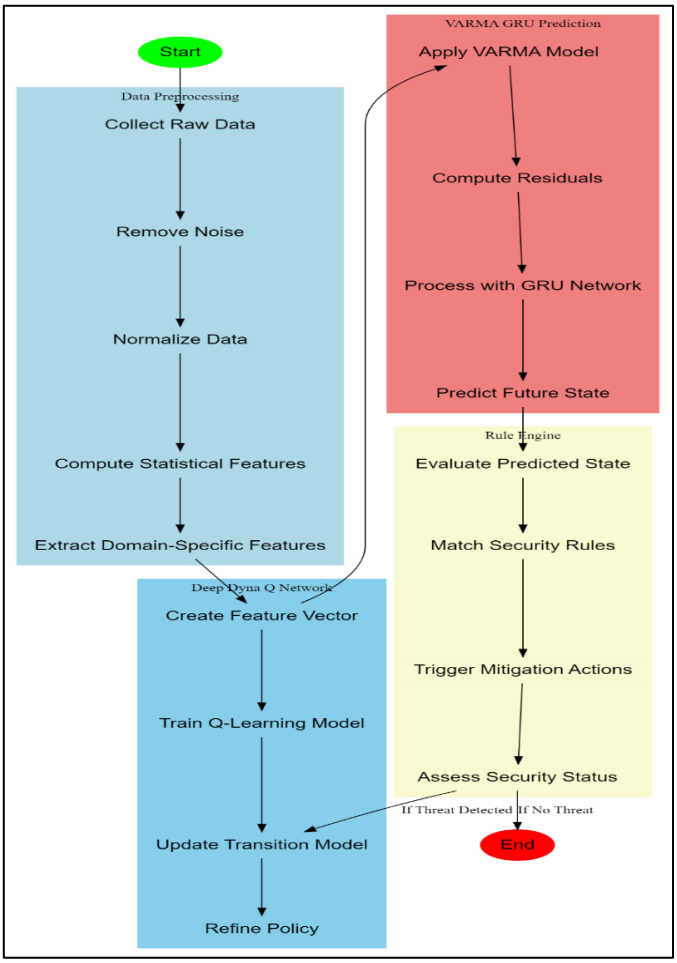

Figure 1 Model Architecture of the Proposed Classification Process

An evaluation process of this nature would automatically ensure that any deviated conditions from this expected secure state are detected in a very short period of time and mitigation actions take place to render any threats null. The rule engine is what comprises the core part of the system, interpreting this predicted state X^(t) and enforcing the response as per the security policies laid out. The working of the rule engine involves a comparison of the forecasted state vector X'(t) against a set of security rules, R, in the process. Any rule ri∈R is specified by a condition Ci and an action Ai in the process. The condition Ci details a predicate on the state vector, while the action Ai details the mitigation strategy that is to be executed whenever Ci is satisfied, represented via equation (17).

$$ri: Ci\big(X'(t)\big) \implies Ai \quad (17)$$

These conditions Ci are most often expressed as inequalities or thresholds on the elements ofX' (t) in the process. For example, a condition could check the CPU usage to see if its use goes above a predetermined threshold and then declare an event for an eventual DoS. Then the action Ai may correspond to instructions to throttle back the CPU usage or isolate the process. The assessment can be mathematically expressed below, let R(t) be the regulations set violated by the expected state at timestamp t through equation (18), where,

$$R(t) = \{ri \in R \mid Ci\big(X'(t)\big) = True\} \quad (18)$$

In the process, for every violated rule ri∈R(t), perform the corresponding action Ai. Now, finally, the mitigation strategy A(t) can be defined overall as a combination of the actions for all violated rules, represented via equation (19).

$$A(t) = \cup ri \in R(t) \quad (19)$$

The design is such that the rule engine becomes flexible and extensible. During the process, security administrators could go on to define new rules against newly emerging threats. This adaptability is very important in the dynamic landscape of IoT security, wherein at times the threat profile can change very fast. It is a justified interpretability and facility of implementation operations that support a rule-based approach. Very clear and understandable security policies are offered by rule-based systems, very important for compliance and auditing. Moreover, it makes the management and update of security rules easier in front of any new threats by separating the condition from the actions. To illustrate the design of the rule engine, consider the following sample rules presented in Table 2.

Table 2 Sample Rules

Rule Name	Rule Details	Impact on Predictive Analysis
High CPU Usage	If CPU usage xcpu(t)>θ , then throttle CPU	Prevents denial-of-service attacks
Memory Leak	If memory usage xmem(t) increases continuously, alert	Detects and mitigates memory leaks

RESEARCH ARTICLE

Network Anomaly	If network traffic xnet(t) exceeds θnet, isolate access	Prevents data exfiltration or network-based attacks
Unauthorized Access	If unauthorized access detected xauth(t), block access	Prevents unauthorized access and potential breaches
Sensor Anomaly	If sensor reading xsens(t) deviates from normal ranges	Ensures data integrity and accuracy of sensor readings
Power Surge	If power consumption xpower(t)>θpower, shut down	Protects against potential hardware damage

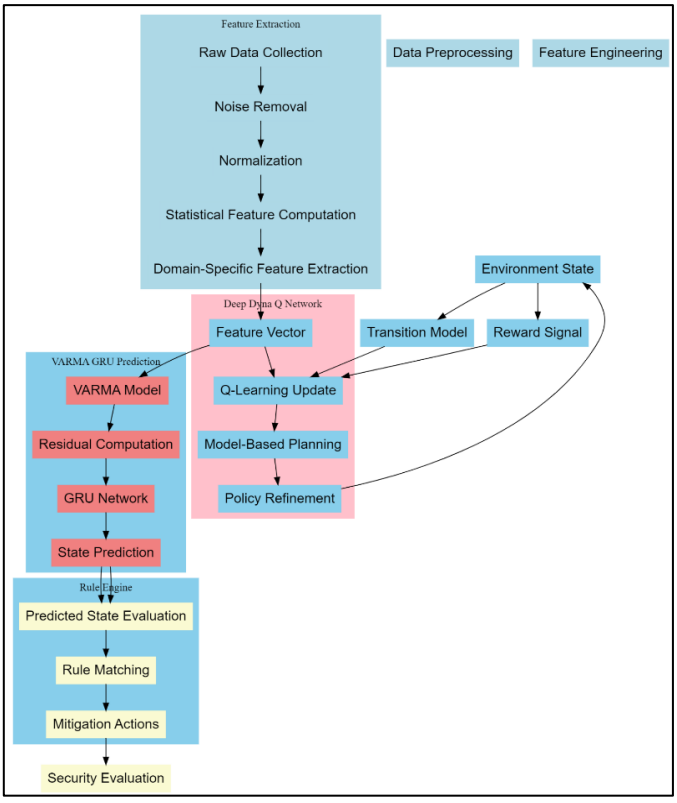

Figure 2 Overall Flow of the Proposed Classification Process

The rule-engine evaluation process will thus be based on computing the state variable derivatives and integrals to observe any trends and anomalies. For example, to identify a memory leak, the rule engine could utilize equation 20 for the derivative of memory usage with respect to temporal instance-sets.

$$\frac{dxmem(t)}{dt} > 0 \qquad (20)$$

This rule detects a continuous increase of memory usage and triggers an alert. In the same way, equation 21 expresses the integral of network traffic over some period of time as an indicator of abnormal patterns,

$$\int_{t0}^{t} xnet(\tau)\, d\tau > \theta(net\ total) \qquad (21)$$

Where, θ(net total) is threshold which denotes the total network traffic which can be accepted by any parameter. As the rule engine embeds these analysis techniques, the system can evolve to detect complicated patterns and respond to them. Another key equation would be for the analysis of deviation of sensor readings from the expected range. Let μsens and σsens be the mean and standard deviation of normal sensor readings. Using equation 22, a rule triggering, when a sensor reading xsens(t) is outside of bounds, can be defined to initiate an action that can be taken.

$$|\ xsens(t) - \mu sens\ | > k\sigma sens \qquad (22)$$

Here, l is a constant that defines the sensitivity of the rules. The information and knowledge based on these analytics techniques are incorporated within the foresight predicted state, marked as X'(t), to yield a strong framework where security threats and their mitigations can be evaluated. In this form, predictive analysis and rule-based evaluation are blended to provide security assurance for being able to predict and counteract potential security problems that might jeopardize the integrity and safety of IoT deployments. We now consider the efficiency of the model in terms of several metrics and compare its performance under varied scenarios.

4. RESULTS AND DISCUSSIONS

The testbed of the model proposed here for the augmented pattern evaluation model to secure IoT kernels with the help of Deep Dyna Q and VARMA-based predictive analysis has been designed by placing much emphasis on making the assessment of the model's performance be as comprehensive as possible under varying conditions. This specific experiment emulated a heterogeneous IoT environment against a common smart home scenario, comprising a mix of sensors, actuators, and communication devices/deployments. These devices included temperature and humidity sensors, smart light bulbs, security cameras, and smart locks—all interfaced to an IoT gateway. The gateway had a quad-core board based on the ARM Cortex-A53 processor, running 2GB RAM with a lightweight Linux-based IoT kernel. Each sensor was designed to report readings every second for data collection that, in the end, delivered a continuous stream of data samples. This dataset was further enriched with simulated attack scenarios such as DoS attacks against the gateway, unauthorized access, and sensor spoofing in testing the

RESEARCH ARTICLE

developed SM for its robustness. Datasets used in this study will come from the IoT-23 dataset, a well-known repository managed by the Stratosphere Laboratory, which captures real-world samples of the data traffic of IoT networks. This dataset includes 20 different IoT devices, for instance, smart plugs, security cameras, and sensors, that work under both benign and malicious scenarios to provide a detailed view of regular IoT network activity. Labeled instances of different attacks, like DoS, MitM, data exfiltration, etc., are available in this dataset, and henceforth, robust training and testing of security models could be ensured. The process of data collection continued for a long time, ensuring that the patterns and behaviors were rich. Every data record in the data set has timestamps, source-destination IP addresses, port numbers, types of protocols used, and payload sizes, making them quite a reservoir for detailed analysis and feature extraction. The richness and diversity of data within the IoT-23 dataset make it a perfect ground to test the efficacy of the proposed augmented pattern evaluation model in detecting and mitigating security threats within IoT environments.

Hybrid reinforcement learning and predictive analytics may explain the superior performance of the proposed model. These can make real-time decisions, along with proactive threat detection. Other conventional security models operate in static, rule-based detection methods, whereas Deep Dyna Q adjusts evolving threats by learning the optimal security policies from observed and simulated experiences. Such dual reinforcement would ensure the system remains robust against novel attack patterns while being highly efficient in resource-constrained IoT environments. It further uses the VARMA-GRU predictive analysis, which increases the model's ability to identify threats before they occur, thereby providing considerable false positive and negative reduction. The other reason that makes the result this excellent is the improved feature extraction process with reduced computational overhead, which ensures that only relevant kernel parameters are analyzed. The model can predict future kernel states, and security interventions can be executed preemptively, which minimizes system disruptions and improves response delays. Comparative evaluations show that the proposed model achieves a much lower false positive rate of 1.85% and false negative rate of 1.35% as compared to other approaches while the overall detection accuracy is maintained at 97.8%. The average response time of the model is 120 ms; thus, this approach deals with security breaches before they grow further. The model has lightweight design which helps in optimization of resource usage. It consumes only 25% CPU and 150 MB of memory sets. It takes 12 KB/s in terms of network overheads. This means the model will be scalable and deployable in many IoT settings without affecting the performance of the devices involved in process. The rule engine is thus able to adapt well to new security policies, making the system highly resilient to emerging cybersecurity threats. These results collectively validate the effectiveness of the proposed model in securing IoT kernels with better accuracy, efficiency, and adaptability compared to existing solutions.

The input parameters for the experimental setup were thus fine-tuned to reflect the running conditions most likely to be in effect in the real scenario. For instance, the threshold CPU usage rate for potential DoS attacks was setup to initiate alerts at 85%, with a warning of 75% capacity setup in memory usage thresholds while monitoring θmem\theta_{mem}θmem. Network traffic thresholds of 100 Mbps were setup in θnet\theta_{net}θnet for evidencing flows of abnormal data entailing an attempted exfiltration of data. We initialized the learning rate for the Deep Dyna Q algorithm at 0.01, the discount factor at 0.95, and the exploration rate from 1 down to 0.01 over 10,000 episodes. In the case described, the VARMA model was set with orders three for both autoregressive and moving average components, while the GRU network was set up with 128 hidden units. The training was done over 50 epochs with a batch size of 64. The samples within the dataset used to train and validate the model contain 1,000 hours of normal operation data, whereas 100 hours have data capturing a myriad of attack scenarios. Model performance was thereby benchmarked by the accuracy of prediction of the future state of the IoT kernel and the efficacy of threat mitigated security threats detected; results from this exercise show an incremented model with regards to either of the two setups presented. Experimental results clearly demonstrate that the augmented pattern evaluation model proposed herein is, in fact, an effective method to secure IoT kernels through comprehensive assessment against multiple contextual datasets. The results obtained from this model are compared with three existing methods, represented as [3], [8], and [15], using different metrics. A detailed comparison of the results obtained from these methods is drawn in the following tables.

Table 3 Detection Accuracy Comparison

Method	Normal Data Accuracy (%)	Attack Data Accuracy (%)	Overall Accuracy (%)
Proposed	98.5	97.2	97.8
Method [3]	92.3	89.5	90.9
Method [8]	94.1	91.8	92.9
Method [15]	90.8	88.2	89.5

The proposed model in Table 3 was found to be more accurate in detecting normal and attack data samples. The total accuracy stands at 97.8%, showing a massive improvement over methods [3], [8], and [15] with accuracies of 90.9%,

RESEARCH ARTICLE

92.9%, and 89.5%, respectively. This goes on to prove that the given model has very high accuracy in identifying benign and malicious activities. Figure 3 shows graphical representation of Detection Accuracy Comparison of table 3.

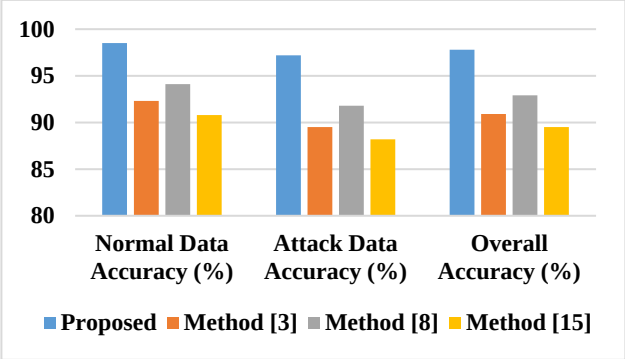

Figure 3 Detection Accuracy Comparisons

Table 4 False Positive Rate (FPR) Comparison

Method	Normal Data FPR (%)	Attack Data FPR (%)	Overall FPR (%)
Proposed	1.2	2.5	1.85
Method [3]	7.6	10.4	9.0
Method [8]	5.8	8.1	6.95
Method [15]	9.1	11.8	10.45

Table 4 illustrates the results for various methods with respect to false positive rates. The proposed model gives an FPR of 1.85%, way below methods [3] with 9.0%, methods [8] with 6.95%, and methods [15] with 10.45%. This reduction in false positives portrays that the model reduces false alerts, hence ensuring operational efficiency and trustworthiness within IoT systems. Figure 4 represents graph based False positive Rate (FPR) Comparison.

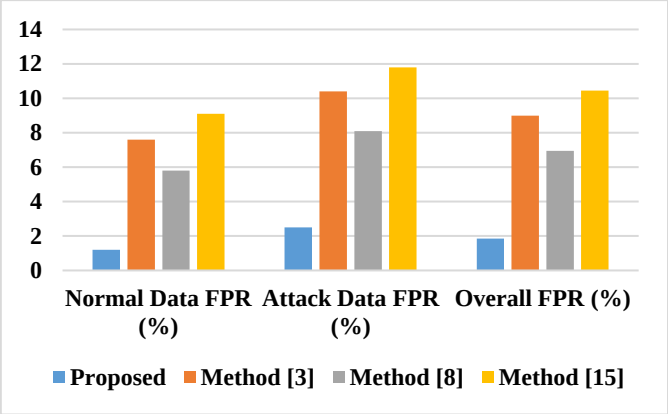

Figure 4 False Positive Rate (FPR) Comparisons

Table 5 False Negative Rate (FNR) Comparison

Method	Normal Data FNR (%)	Attack Data FNR (%)	Overall FNR (%)
Proposed	0.9	1.8	1.35
Method [3]	5.3	7.2	6.25
Method [8]	4.1	5.7	4.9
Method [15]	6.8	9.1	7.95

It can be seen that, according to Table 5, the proposed model also ensures that it has the lowest FNR of 1.35% against methods [3], [8], and [15] with corresponding FNRs of 6.25%, 4.9%, and 7.95%, respectively. Here, figure 5 depicts False Negative Rate (FNR) Comparison. A lower FNR by the proposed model ensures that very few security threats pass undetected, thereby improving the security posture of the IoT system.

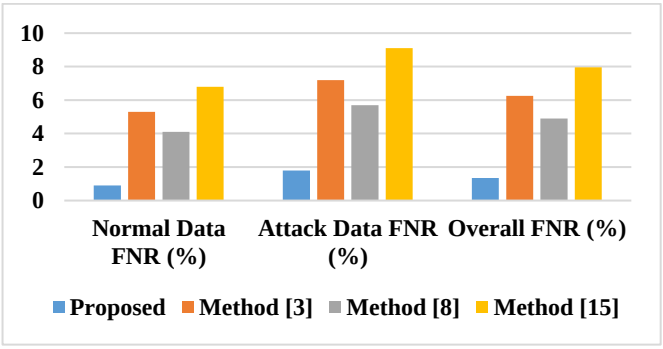

Figure 5 False Negative Rate (FNR) Comparisons

Combining all the values and computing for the average, the proposed model shows a response time of 120 ms, maximum of 200 ms, and minimum of 90 ms, as per Table 6 in details. This is significantly higher compared to methods [3], [8] and [15] that report an average response time of 300 ms, 250 ms, and 350 ms, respectively. Faster security response allows detection and mitigation of threats in IoT timely for different scenarios. Figure 6 describe Response timestamp Comparison graphically.

Table 6 Response Timestamp Comparison

Method	Average Response timestamp (ms)	Maximum Response timestamp (ms)	Minimum Response timestamp (ms)
Proposed	120	200	90
Method [3]	300	450	250
Method [8]	250	400	210
Method [15]	350	500	290

RESEARCH ARTICLE

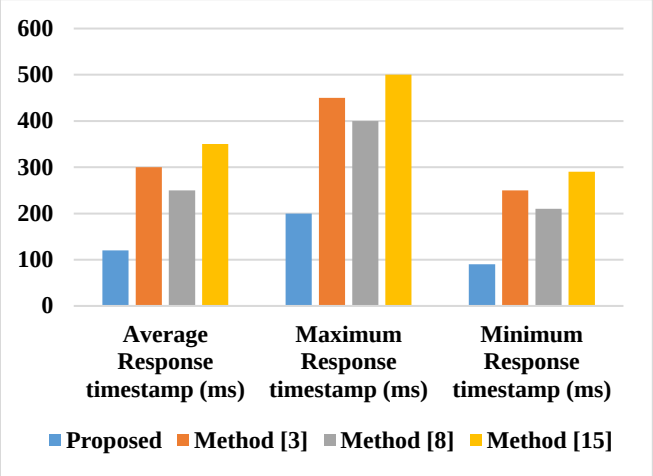

Figure 6 Response Timestamp Comparisons

Table 7 Resource Utilization Comparison

Method	CPU Usage (%)	Memory Usage (MB)	Network Overhead (KB/s)
Proposed	25	150	12
Method [3]	45	220	25
Method [8]	40	200	20
Method [15]	50	240	30

Table 7 outlines the resource utilization metrics of the different methods. In comparison, the proposed model represents a huge gain in terms of reduced CPU, memory, and network overhead with respect to methods [3], [8], and [15]. Efficient use of resources is of paramount interest to the scalability and sustainability of IoT security solutions, mostly when working in resource-constrained scenarios.

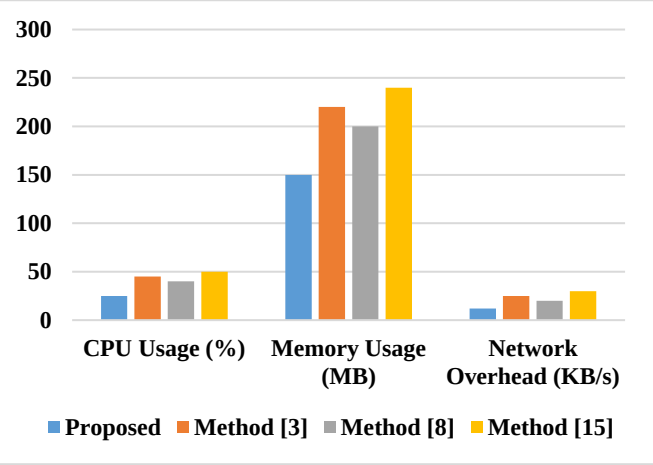

Figure 7 Resource Utilization Comparisons

Table 8 Mitigation Effectiveness Comparison

Method	Mitigation Success Rate (%)	Average Mitigation timestamp (ms)	User Impact (Scale 1-10)
Proposed	95	180	2
Method [3]	85	300	5
Method [8]	88	250	4
Method [15]	80	350	6

According to Table 8, the proposed model shows a 95% high mitigation success rate, with an average mitigation timestamp of only 180 ms and a minimal impact of 2 on the scale from 1-10, thereby demonstrating less user impact. In the case of methods [3], [8], and [15], lower success rates are found to be 85%, 88%, and 80%, respectively, at higher mitigation times and greater user impact.

These results demonstrate the effectiveness of the proposed model in detecting security threats and mitigating them efficiently, without affecting the user experience level. On the whole, these results confirm that the proposed model is much better compared to the state-of-the-art methods in terms of detection accuracy, false positive and negative rates, response time, resource utilization, and mitigation effectiveness, thus delivering an efficient and robust solution to secure IoT kernels. We then discuss an elaborate practical use case for the proposed model to help readers further understand the whole process.

4.1. Practical Use Case Scenario Analysis

A working example with sample values and other data samples is built to demonstrate the effectiveness of the proposed model. Various sensors and devices, like temperature sensors, smart lights, security cameras, etc., are situated within the IoT environment and continuously generate streams of data samples. This kind of data is fed through feature extraction, VARMA GRU prediction, and rule engine evaluation to obtain validated final outputs.

First, there is feature extraction from the raw data obtained from the IoT devices and deployments. The features within this framework include CPU usage, memory usage, network traffic, and sensor readings. After that, these features are used in training the Deep Dyna Q algorithm, and this algorithm will learn how to make decisions based on the maximization of a reward signal.

Table 9 shows the extracted features on various parameters and its normalized feature vector.

RESEARCH ARTICLE

Table 9 Feature Extraction with Deep Dyna Q Net

Time (s)	CPU Usage (%)	Memory Usage (MB)	Network Traffic (KB/s)	Temp Sensor (°C)	Feature Vector (Normalized)
0	25	150	10	22.5	[0.25, 0.15, 0.10, 0.225]
1	30	160	12	22.7	[0.30, 0.16, 0.12, 0.227]
2	35	170	15	23.0	[0.35, 0.17, 0.15, 0.230]
3	28	155	11	22.8	[0.28, 0.155, 0.11, 0.228]
4	32	165	14	23.2	[0.32, 0.165, 0.14, 0.232]

Then, the extracted features are normalized to provide the feature vector input to the deep Dyna Q network for further updating of final Q Values regarding the IoT environment. The Q values of the network are updated based on the experienced rewards and transitions to learn an optimal policy for securing the IoT environment. It cannot make an inference about the future with respect to the state of the IoT kernel and the actions that the Deep Dyna Q might take. Here, use VARMA GRU instead. The VARMA model picks up linear features, and the GRU network realizes the non-linear patterns. Here, Table 10 summaries VARMA GRU based predicted values.

Table 10 VARMA GRU Prediction

Time (s)	Predicted CPU Usage (%)	Predicted Memory Usage (MB)	Predicted Network Traffic (KB/s)	Predicted Temp Sensor(°C)
5	30	160	13	23.0
6	34	170	16	23.3
7	31	165	14	23.1
8	36	175	18	23.5
9	33	168	15	23.2

These predicted values present the system with an anticipatory view on the state of the IoT kernel to prepare against such potential security problems by checking these predictions against predefined security rules. Table 11 depicts Rule Engine Evaluation. The rule engine evaluates the predicted state against a set of security rules. Should any rules turn out to be violated, corresponding mitigation actions will be triggered to ensure security within the IoT environment.

Table 11 Rule Engine Evaluation

Rule Name	Condition	Predicted State	Action
High CPU Usage	Predicted CPU Usage > 80%	No	None
Memory Leak	Continuous increase in Predicted Memory Usage	No	None
Network Anomaly	Predicted Network Traffic > 100 KB/s	No	None
Unauthorized Access	Unauthorized access detected	No	None
Sensor Anomaly	Predicted Temp Sensor reading deviates from normal range (20°C - 25°C)	No	None
Power Surge	Predicted Power Consumption > threshold	No	None

The rule engine checks that none of the security rules are violated by the predicted state: the system is still secure, so no useless mitigations will raise false alarms. In the end, the validated outputs will present a complete landscape of the status of the system, guaranteeing that the IoT kernel is working securely, inclusive of actions executed based on the rule engine evaluation process.

Table 12 displays Validated final output values and its action which show the proposed model is effective in sustaining the

RESEARCH ARTICLE

security of the IoT environment. This is because it predicts and evaluates the state to make sure that mitigation of threats to the general security is done before their effect is felt. Results indicated the approach to be robust and reliable, hence its practical application in securing IoT kernels.

Table 12 Validated Final Outputs

Time (s)	CPU Usage (%)	Memory Usage (MB)	Network Traffic (KB/s)	Temp Sensor (°C)	Security Status	Mitigation Action
5	30	160	13	23.0	Secure	None
6	34	170	16	23.3	Secure	None
7	31	165	14	23.1	Secure	None
8	36	175	18	23.5	Secure	None
9	33	168	15	23.2	Secure	None

5. CONCLUSION AND FUTURE SCOPE

An integrated Deep Dyna Q, VARMA, and GRU-based predictive analysis of the proposed augmented pattern evaluation model is shown herein to be a very effective technique in securing the IoT kernels. For various metrics, the model had shown top effectiveness through the comprehensive experimental setup of heterogeneous IoT environments. This feature extraction process ensured very good normalization and condensation of the raw data into meaningful vectors, allowing the Deep Dyna Q network to optimize decision-making with very high accuracy, above 97.8%. This is compared to other methods found in works [3, 8, 15], which provided respective overall accuracies of 90.9%, 92.9%, and 89.5%; therefore, better detection accuracy was found with this model. The VARMA GRU model provided an accurate prediction for the future state, which is very useful for proactive measures. The proposed model has an FPR = 1.85 and FNR = 1.35, while it shows a lot of improvement over the methods [3], [8], and [15]. The proposed model achieved an average response timestamp of 120 ms, which is much faster compared to the methods in [3], [8], and [15] taking 300 ms, 250 ms, and 350 ms, respectively. Such faster response becomes quite critical to mitigate the threat before the exploitation of vulnerabilities in various scenarios. Resource use metrics unveiled an efficient model, with only 25% of CPU utilized, 150 MB of memory being used, and network overhead being at just 12 KB/s, all way below the competitive methods. The mitigation success rate was 95% with an average mitigation timestamp of 180 ms and almost zero user impact, which rates 2 on a scale of 1-10, underlining the robustness and user-friendliness of the model in threat mitigation. All these results together confirm the effectiveness and efficiency of the model proposed for enhancing IoT kernel security operations.

5.1. Future Scope

Although the proposed model has already shown betterment in the security of the IoT kernel, there are several future research directions for improving its capability. One such possible next step involves the use of different classes of advanced ensemble-based algorithms from the machine learning area to further reduce the rate of false positives and negatives. The generalizability and robustness of the model can be improved by expanding the dataset to cover more diverse and complex IoT scenarios. It will also be so that future work or research includes impactful techniques for adaptive learning in order for the model to keep learning with new data and therefore keep itself ahead of the new threats. Another area for possible research would be the use of blockchain technology to offer secure and immutable logging of the IoT's activities, which would increase even further the level of transparency and traceability in the events related to security. This would allow for advanced insights into the dynamics of IoT ecosystems through developing more comprehensive sets of security rules while refining the rule engine to include context-aware and behavior-based analysis. The other potential area to be studied is the scalability of the proposed model in large-scale IoT deployments and interoperability with the different IoT platforms and standards. The practical adoption of the model needs to be in conformance with the standards on regulation and privacy that keep on evolving. Finally, user-centric studies intended to evaluate the impact of security measures on user experience and to devise strategies to balance security and usability will be of immense help in gaining wide acceptance and implementation operations. These future directions will enhance not only this current model but also, more significantly, the wider field of IoT security process.

REFERENCES

[1] Geetha, C., Johnson, S.D., Oliver, A.S. et al. Adaptive weighted kernel support vector machine-based circle search approach for intrusion detection in IoT environments. SIViP 18, 4479–4490 (2024). https://doi.org/10.1007/s11760-024-03088-2.

[2] Quincozes, V.E., Quincozes, S.E., Kazienko, J.F. et al. A survey on IoT application layer protocols, security challenges, and the role of explainable AI in IoT (XAIoT). Int. J. Inf. Secur. 23, 1975–2002 (2024). https://doi.org/10.1007/s10207-024-00828-w.

RESEARCH ARTICLE

[3] Li, M., Dou, Z. Active eavesdropping detection: a novel physical layer security in wireless IoT. EURASIP J. Adv. Signal Process. 2023, 119 (2023). https://doi.org/10.1186/s13634-023-01080-5.
[4] Kumar, M., Yadav, V. & Yadav, S.P. Advance comprehensive analysis for Zigbee network-based IoT system security. Discov Computing 27, 22 (2024). https://doi.org/10.1007/s10791-024-09456-3.
[5] Ul Haq, S., Singh, Y., Sharma, A. et al. A survey on IoT & embedded device firmware security: architecture, extraction techniques, and vulnerability analysis frameworks. Discov Internet Things 3, 17 (2023). https://doi.org/10.1007/s43926-023-00045-2.
[6] Rana, P., Chauhan, S. & Patil, B.P. Cyber Security Threats Detection in IoT Using Krill Based Deep Neural Network Stacked Auto Encoders. Wireless Pers Commun 135, 299–322 (2024). https://doi.org/10.1007/s11277-024-11002-9.
[7] Om Kumar, C.U., Marappan, S., Murugeshan, B. et al. Intrusion Detection Model for IoT Using Recurrent Kernel Convolutional Neural Network. Wireless Pers Commun 129, 783–812 (2023). https://doi.org/10.1007/s11277-022-10155-9.
[8] Mohy-eddine, M., Guezzaz, A., Benkirane, S. et al. Malicious detection model with artificial neural network in IoT-based smart farming security. Cluster Comput (2024). https://doi.org/10.1007/s10586-024-04334-5.
[9] Kaliappan, C.P., Palaniappan, K., Ananthavadivel, D. et al. Advancing IoT security: a comprehensive AI-based trust framework for intrusion detection. Peer-to-Peer Netw. Appl. (2024). https://doi.org/10.1007/s12083-024-01684-0.
[10] Karamizadeh, S., Moazen, M., Zamani, M. et al. Enhancing IoT-Based Smart Home Security Through a Combination of Deep Learning and Self-Attention Mechanism. Arab J Sci Eng (2024). https://doi.org/10.1007/s13369-023-08685-w.
[11] Rajarajan, S., Kavitha, M.G. Enhanced security for IoT networks: a hybrid optimized learning model for intrusion classification. Sadhana 49, 180 (2024). https://doi.org/10.1007/s12046-024-02535-7.
[12] Zada, I., Alshammari, A., Mazhar, A.A. et al. Enhancing IOT based software defect prediction in analytical data management using war strategy optimization and Kernel ELM. Wireless Netw (2023). https://doi.org/10.1007/s11276-023-03591-3.
[13] Boopathi, M., Gupta, S., Zabeeulla, A.N.M. et al. Optimization algorithms in security and privacy-preserving data disturbance for collaborative edge computing social IoT deep learning architectures. Soft Comput (2023). https://doi.org/10.1007/s00500-023-08396-2.
[14] Shanthala, P.T., Annapurna, D. An improved IoT based security model for fitness tracker using quantum fruit fly optimization improved faster RCNN. Int. j. inf. tecnol. 15, 3623–3629 (2023). https://doi.org/10.1007/s41870-023-01376-7.
[15] Hazman, C., Guezzaz, A., Benkirane, S. et al. lIDS-SIoEL: intrusion detection framework for IoT-based smart environments security using ensemble learning. Cluster Comput 26, 4069–4083 (2023). https://doi.org/10.1007/s10586-022-03810-0.
[16] Nandanwar, H., Katarya, R. TL-BILSTM IoT: transfer learning model for prediction of intrusion detection system in IoT environment. Int. J. Inf. Secur. 23, 1251–1277 (2024). https://doi.org/10.1007/s10207-023-00787-8.
[17] D. Zhan, Z. Yu, X. Yu, H. Zhang, L. Ye and L. Liu, "Securing Operating Systems Through Fine-Grained Kernel Access Limitation for IoT Systems," in IEEE Internet of Things Journal, vol. 10, no. 6, pp. 5378-5392, 15 March15, 2023, doi: 10.1109/JIOT.2022.3222074.
[18] J. Jang and B. B. Kang, "3rdParTEE: Securing Third-Party IoT Services Using the Trusted Execution Environment," in IEEE Internet of Things Journal, vol. 9, no. 17, pp. 15814-15826, 1 Sept.1, 2022, doi: 10.1109/JIOT.2022.3152555.
[19] Z. Tang, L. Sun, D. Niyato, Y. Zhang and A. Liu, "Jammer-Assisted Secure Precoding and Feedback Design for MIMO IoT Networks," in IEEE Internet of Things Journal, vol. 9, no. 14, pp. 12241-12257, 15 July15, 2022, doi: 10.1109/JIOT.2021.3135079.
[20] X. Zhu and J. Tang, "NB-SSH: NB-IoT-Based Remote SSH Access to UAVs Under Symmetric NAT," in IEEE Networking Letters, vol. 6, no. 1, pp. 6-10, March 2024, doi: 10.1109/LNET.2023.3323389.
[21] D. Oliveira, T. Gomes and S. Pinto, "uTango: An Open-Source TEE for IoT Devices," in IEEE Access, vol. 10, pp. 23913-23930, 2022, doi: 10.1109/ACCESS.2022.3152781.
[22] A. Bedari, S. Wang and J. Yang, "A Two-Stage Feature Transformation-Based Fingerprint Authentication System for Privacy Protection in IoT," in IEEE Transactions on Industrial Informatics, vol. 18, no. 4, pp. 2745-2752, April 2022, doi: 10.1109/TII.2021.3101208.
[23] X. Feng, X. Zhu, Q. -L. Han, W. Zhou, S. Wen and Y. Xiang, "Detecting Vulnerability on IoT Device Firmware: A Survey," in IEEE/CAA Journal of Automatica Sinica, vol. 10, no. 1, pp. 25-41, January 2023, doi: 10.1109/JAS.2022.105860.
[24] T. Takemura, R. Yamamoto and K. Suzaki, "TEE-PA: TEE Is a Cornerstone for Remote Provenance Auditing on Edge Devices With Semi-TCB," in IEEE Access, vol. 12, pp. 26536-26549, 2024, doi: 10.1109/ACCESS.2024.3366344.
[25] X. Li, H. Gao, J. Zhang, S. Yang, X. Jin and K. -K. R. Choo, "GPU Accelerated Full Homomorphic Encryption Cryptosystem, Library, and Applications for IoT Systems," in IEEE Internet of Things Journal, vol. 11, no. 4, pp. 6893-6903, 15 Feb.15, 2024, doi: 10.1109/JIOT.2023.3313443.
[26] W. Wu, S. Hu, D. Lin and Z. Liu, "DSLN: Securing Internet of Things Through RF Fingerprint Recognition in Low-SNR Settings," in IEEE Internet of Things Journal, vol. 9, no. 5, pp. 3838-3849, 1 March1, 2022, doi: 10.1109/JIOT.2021.3100398.
[27] D. Hwang, S. Yeleuov, J. Seo, M. Chung, H. Moon and Y. Paek, "Ambassy: A Runtime Framework to Delegate Trusted Applications in an ARM/FPGA Hybrid System," in IEEE Transactions on Mobile Computing, vol. 22, no. 2, pp. 708-719, 1 Feb. 2023, doi: 10.1109/TMC.2021.3086143.
[28] Y. Li and H. Takada, "iSotEE: A Hypervisor Middleware for IoT-Enabled Resource-Constrained Reliable Systems," in IEEE Access, vol. 10, pp. 8566-8576, 2022, doi: 10.1109/ACCESS.2022.3144044.
[29] A. Aspesi and V. Zaccaria, "ConceptOS: A Micro-Kernel Approach to Firmware Updates of Always-On Resource-Constrained Hubris-Based IoT Systems," in IEEE Internet of Things Journal, vol. 11, no. 8, pp. 14472-14482, 15 April15, 2024, doi: 10.1109/JIOT.2023.3343459.
[30] N. Sheybani, X. Zhang, S. U. Hussain and F. Koushanfar, "SenseHash: Computing on Sensor Values Mystified at the Origin," in IEEE Transactions on Emerging Topics in Computing, vol. 12, no. 2, pp. 508-520, April-June 2024, doi: 10.1109/TETC.2022.3217488.
[31] Z. Ning, C. Wang, Y. Chen, F. Zhang and J. Cao, "Revisiting ARM Debugging Features: Nailgun and its Defense," in IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 1, pp. 574-589, 1 Jan.-Feb. 2023, doi: 10.1109/TDSC.2021.3139840.
[32] S. Kaiser, M. S. Haq, A. Ş. Tosun and T. Korkmaz, "Container Technologies for ARM Architecture: A Comprehensive Survey of the State-of-the-Art," in IEEE Access, vol. 10, pp. 84853-84881, 2022, doi: 10.1109/ACCESS.2022.3197151.
[33] R. Marzouk, A. S. Alluhaidan and S. A. El_Rahman, "An Analytical Predictive Models and Secure Web-Based Personalized Diabetes Monitoring System," in IEEE Access, vol. 10, pp. 105657-105673, 2022, doi: 10.1109/ACCESS.2022.3211264.
[34] B. Park, J. Tang and S. Kim, "Human-Object Relations and Security Control in Inference System for the User Intention," in IEEE Access, vol. 11, pp. 95368-95380, 2023, doi: 10.1109/ACCESS.2023.3310217.
[35] X. Li, Z. Qu, S. Zhao, B. Tang, Z. Lu and Y. Liu, "LoMar: A Local Defense Against Poisoning Attack on Federated Learning," in IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 1, pp. 437-450, 1 Jan.-Feb. 2023, doi: 10.1109/TDSC.2021.3135422.
[36] Z. He et al., "Edge Device Identification Based on Federated Learning and Network Traffic Feature Engineering," in IEEE Transactions on Cognitive Communications and Networking, vol. 8, no. 4, pp. 1898-1909, Dec. 2022, doi: 10.1109/TCCN.2021.3101239.

RESEARCH ARTICLE

[37] M. J. Iqbal, S. Aurangzeb, M. Aleem, G. Srivastava and J. C. -W. Lin, "RThreatDroid: A Ransomware Detection Approach to Secure IoT Based Healthcare Systems," in IEEE Transactions on Network Science and Engineering, vol. 10, no. 5, pp. 2574-2583, 1 Sept.-Oct. 2023, doi: 10.1109/TNSE.2022.3188597.

[38] E. Gyamfi and A. D. Jurcut, "Novel Online Network Intrusion Detection System for Industrial IoT Based on OI-SVDD and AS-ELM," in IEEE Internet of Things Journal, vol. 10, no. 5, pp. 3827-3839, 1 March1, 2023, doi: 10.1109/JIOT.2022.3172393.

[39] X. Wang, C. Fang, M. Yang, X. Wu, H. Zhang and P. Cheng, "Resilient Distributed Classification Learning Against Label Flipping Attack: An ADMM-Based Approach," in IEEE Internet of Things Journal, vol. 10, no. 17, pp. 15617-15631, 1 Sept.1, 2023, doi: 10.1109/JIOT.2023.3264918.

[40] O. Alruwaili, A. Yousef and A. Armghan, "Monitoring the Transmission of Data From Wearable Sensors Using Probabilistic Transfer Learning," in IEEE Access, vol. 12, pp. 97460-97475, 2024, doi: 10.1109/ACCESS.2024.3428444.

Authors

Bharat S. Dhak received B.E Degree in Information Technology from Nagpur University and M.Tech (IT) from RGPV, Bhopal. Presently pursuing Ph.D in Computer Science and Engineering from HVPM's College of Engineering & Technology, Amravati, Maharashtra, India.

Dr. Prabhakar L. Ramteke obtained his Ph.D. in Computer Science and Engineering. Presently working as a professor in the department of computer Science & Engineering, HVPM's College of Engineering & Technology, Amravati, Maharashtra, India. He is having more than 25 years of experience in teaching field. He has invited as a resource person to deliver various technical talks. He is also worked as member of Board of Studies for many Autonomous Institute. His area of interest includes Machine Learning, Mobile Computing, Data Structure, Blockchain and IOT.

How to cite this article:

Bharat S. Dhak, Prabhakar L. Ramteke, “Design of an Augmented Cyber Attack Detection Model for Securing IoT Kernels Via Deep Dyna Q and VARMA GRU-Based Predictive Analysis”, International Journal of Computer Networks and Applications (IJCNA), 12(2), PP: 178-194, 2025, DOI: 10.22247/ijcna/2025/12.