International Journal of Scientific Research in Computer Science and Engineering
Vol.**12**, Issue.**6**, pp.**14-24**, December **2024**
E-ISSN: 2320-7639
Available online at: www.isroset.org

Research Article

A Reinforcement Learning-Based Multi-Agent System for Advanced Network Attack Prediction

Stephen Kahara Wanjau[1*], **Stephen Njenga Thiiru**[2]

[1,2]Dept. of Computer Science, Murang'a University of Technology, Murang'a, Kenya

**Corresponding Author: steve.kahara@gmail.com*

Received: 18/Oct/2024; **Accepted**: 20/Nov/2024; **Published**: 31/Dec/2024

Abstract— This paper addresses the challenges of traditional Network Intrusion Detection Systems (NIDS) in handling the increasing complexity and volume of modern cyberattacks. The authors suggest a novel multi-agent deep reinforcement learning (MADRL) approach, employing a deep Q-network (DQN) architecture with convolutional and fully connected layers. This architecture incorporates Target networks and Experience Replay to enhance learning and adaptation. A hierarchical reinforcement learning strategy decomposes complex intrusion detection tasks into manageable subtasks, enabling efficient exploration of high-dimensional state-action spaces. The proposed model, trained and evaluated on the CICIDS2017 dataset using a 70% training set and 30% test split and 10-fold cross-validation, achieves exceptional performance. It attains 97.71% accuracy, 98.34% recall, 97.29% precision, and 96.76% F1-score after 50 iterations, surpassing existing NIDS solutions in comparative analysis. The model's strength lies in its ability to effectively mimic environmental characteristics through multi-agent learning, leading to robust detection of intricate attack patterns. Furthermore, our approach demonstrates strong generalization capabilities on unseen data, indicating its potential for real-world deployment. This research contributes significantly to the evolution of intelligent network security systems by introducing an innovative MADRL framework. Future research directions include implementing the solution in real-time network environments, expanding the agent network, and extending the model's application to outlier detection and software-defined networking. This work lays the foundation for future advancements in cyber threat detection and mitigation, paving the way for more robust and adaptive network security solutions.

Keywords— Network Intrusion Detection Systems, Multi-Agent Systems, Deep Reinforcement Learning, Deep Q-Network, Cybersecurity, Machine Learning

1. Introduction

The domain of computer and information security, now known as cybersecurity, has been central to research since the inception of digital computing. Network security, particularly the detection of attacks on enterprise networks, has become crucial for protecting system confidentiality, integrity, and availability. Although there are numerous cyber-attack prevention schemes, Intrusion Detection Systems (IDS) [1] have become essential tools for identifying malicious network traffic and computer activities that traditional firewalls may overlook. These systems have become essential components in modern network infrastructure for enforcing network security policies.

Intelligent agent technology represents a rapidly expanding research domain, with applications proliferating across industrial and commercial sectors. Multiple studies have explored and developed Multi-Agent Systems (MAS) based intrusion detection solutions to combat threats and protect organizational information assets [2]. These systems leverage agent cooperation and communication, combining their collective knowledge and experience to optimize decision-making. Building on the successful application of multi-agent systems in addressing complex classification problems, this paper proposes a novel approach to enhance network security through MAS-based intrusion detection. The proposed model augments traditional network intrusion detection systems with intelligent agents capable of autonomous decision-making for intrusion detection.

The model is evaluated using the CICIDS2017 benchmark dataset [3], which encompasses contemporary benign activities and malicious attacks representative of modern network traffic.

1.1 Research Motivation

The rapid evolution of cyber threats and the increasing sophistication of network attacks have created significant challenges in maintaining effective network security systems. Conventional intrusion detection methods frequently face challenges with new attack patterns and the growing complexity of contemporary network environments. These challenges are compounded by the rapid rise in network traffic volume, the variety of connected devices, and the introduction of new attack vectors in today's networks. While existing solutions have made progress in threat detection, they frequently face limitations in adapting to emerging threats and handling the high-dimensional nature of network security data. These limitations, coupled with the critical need for real-time threat detection and response capabilities, motivate our research into developing a more robust and adaptive approach to network intrusion detection. Capitalizing on the capabilities of multi-agent systems and deep reinforcement learning, we aimed to create a more resilient and intelligent detection framework capable of evolving with the threat landscape while maintaining high accuracy and minimal false positives.

1.2 Research Objectives

This study was guided by the following research objectives:

i. To develop a multi-agent deep Q-network framework for detecting network intrusions using reinforcement learning architecture.
ii. To implement hierarchical reinforcement learning for decomposing complex intrusion detection tasks into efficient learning components.
iii. To validate the model's efficacy using CICIDS2017 dataset through comprehensive performance metrics analysis.

The remaining part of this work is structured as follows: Section 2 provides an overview of network intrusion detection systems. Section 3 explores intelligent agent theory and multi-agent systems. Section 4 reviews related work on multi-agent applications in network intrusion detection. Section 5 introduces the proposed model. Section 6 outlines the experimental methodology. Section 7 presents and discusses the experimental results. Finally, in section 8 the conclusion is presented and suggestions for future research provided.

2. Network Intrusion Detection Systems: Overview

Intrusion detection serves as a practical approach to enhancing enterprise computer and network security by detecting attacks in real time [4]. Network Intrusion Detection Systems (NIDS) are security applications that monitor and evaluate network traffic, automatically alerting administrators when malicious activities are detected.

The literature discusses three primary forms of intrusion detection systems [5]. First, the Host-based Intrusion Detection Systems (HIDS) operate on specific computers or servers, monitoring activities solely on their host systems. In contrast, Network-based Intrusion Detection Systems (NIDS) examine network traffic across multiple hosts to detect and isolate intrusions, notifying network administrators when suspicious activities arise. Distributed Intrusion Detection Systems (DIDS) employ multiple detection systems interconnected across large networks, either interacting with one another or sending reports to a central server.

Designing effective Network Intrusion Detection Systems (NIDS) presents significant challenges due to the proliferation of novel attacks and diverse network applications. Detection has become increasingly complex due to growing network traffic volumes, widespread encryption, and the expanding ubiquity of modern networks. While Intrusion Detection Systems (IDSs) have attracted substantial research attention from both academia and industry, the challenge of effective detection persists.

The literature classifies intrusion detection approaches into two primary categories: signature-based detection (commonly referred as misuse detection) and anomaly-based detection (also known as behaviour-based detection) [6], [7]. Signature-based detection employs predefined rules or signatures to identify attack patterns, enabling high accuracy and minimal false positives in intrusion identification. In contrast, anomaly-based detection systems identify potential intrusions by comparing actual network behavior against established baseline patterns of normal system activity, flagging deviations as potential security threats.

3. Intelligent Agents Theory

The agent concept has emerged as a fundamental paradigm in computer science, particularly within artificial intelligence, where it guides the design and implementation of intelligent agents. As defined by [8], an intelligent agent is a decision-making entity that operationalizes artificial intelligence, possessing sufficient autonomy to execute specific, predictable, and repetitive tasks for users or applications.

Fox et al. [9] note that the growing sophistication of agent technologies' cognitive capabilities has led to their expanding application across diverse fields. Intelligent agents operate through two primary functions: perception and action. They perceive their environment through sensors and execute actions via actuators or effectors [10]. In effectively accessible environments, agents receive complete sensor information relevant to their goals. In deterministic environments, future states can be predicted from current conditions, eliminating uncertainty [11].

These intelligent agents are structured hierarchically, with sub-agents handling lower-level tasks. The integration of both lower and higher-level agents creates a comprehensive system capable of addressing complex challenges through intelligent behaviors and responses.

Multi-Agent Systems (MAS) represent sophisticated networks of interconnected software agents that collaborate to address challenges exceeding the capabilities or knowledge boundaries of individual agents [12]. These systems have demonstrated particular utility in the design and maintenance of secure networks, offering distributed problem-solving capabilities through autonomous agent interaction.

The fundamental research in Multi-Agent Systems focuses on understanding how autonomous entities (agents) can self-organize to achieve complex tasks and generate emergent phenomena that would be impossible for single agents to accomplish [13], [14]. This collaborative approach enables the system to leverage collective intelligence, where agents share information, coordinate actions, and adapt their behaviours based on interactions with other agents and their environment. The resulting synergy allows MAS to tackle intricate problems through distributed decision-making and parallel processing capabilities.

These systems represent a significant departure from conventional approaches, offering enhanced flexibility and robustness through their distributed architecture. Agents function as sophisticated entities - either physical or logical - operating autonomously on behalf of users across open and distributed environments [15]. Through their actuators, these agents can initiate targeted actions within their operational environment, enabling them to address an expanding range of complex problems. Their ability to perceive, reason, and act independently, while maintaining coordination with other agents, makes MAS particularly suitable in dynamic and challenging domains for instance, network security, where rapid response and adaptive behaviour are crucial. Figure 1 exemplifies multiple agents that interrelates with similar environment.

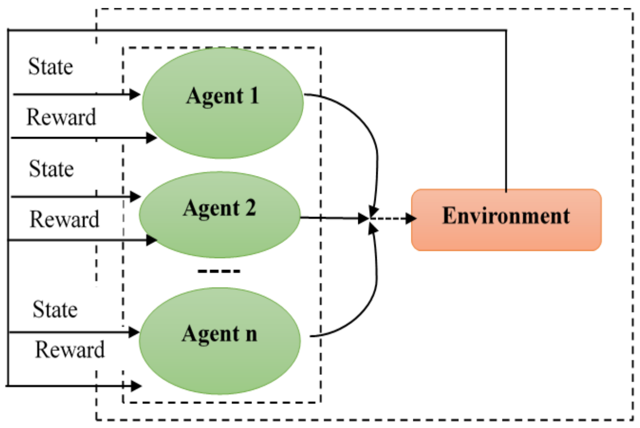

Figure 1. Multiple agents that interact with the same environment (source: [15])

The evolution of multi-agent systems has been significantly advanced through the integration of a single-agent deep reinforcement learning techniques, giving rise to multi agents deep reinforcement learning (MADRL). This emerging field has garnered substantial research attention in recent years [16], [17] as it combines the collaborative power of multi-agent systems with the sophisticated learning capabilities of deep reinforcement learning.

Multi agent systems’ learning process involves a complex and dynamic approach, requiring agents to navigate a high-dimensional state-action space that incorporates the observable states of other agents. While each agent independently determines its actions depending on individual policies, the interconnected nature of the system creates a sophisticated learning environment. The effectiveness of any single agent's behaviour is inherently influenced by and contingent upon the actions and learning progress of other agents within the system [18]. This interdependence creates a rich but challenging learning environment where agents must adapt their strategies not only to achieve their individual objectives but also to respond to the evolving behaviours of other agents in the system.

4. Related Works

The literature presents diverse methodologies for intrusion detection. In this section, a particular emphasis is made on network-based detection systems utilizing multi-agent architectures. Notable among these developments is Liu et al.'s [19] proposal for a Network Intrusion Detection Model grounded in Immune Multi-Agent theory (NIDIMA). The specified model introduces an innovative approach to network protection, demonstrating effectiveness comparable to biological immune systems in network security detection.

A significant contribution to specialized intrusion detection comes from [20], where researchers developed an NIDS specifically for Unmanned Aerial Vehicle (UAV) fleets operating with ad hoc communication architectures. This model addresses security principles of Confidentiality, Integrity, and Authentication in drone networks, which have become increasingly critical across various sectors including e-commerce delivery, agriculture, public safety, energy, and transportation. The model's efficacy was confirmed through the use of the CICIDS2017 dataset, a comprehensive benchmark that incorporates both contemporary attack patterns and normal network activities, providing a robust testing environment for modern intrusion detection systems.

Multi-agent systems have demonstrated significant utility in detecting intrusions particularly in wireless sensor networks. Zhang and Lee [21] pioneered this approach by developing a collaborative intrusion detection system utilizing multi-agent architectures for wireless network security. Expanding on this foundation, Krishnan [22] introduced an efficient distribution technique that enhanced intrusion detection system performance through multi-agent implementation. This self-adaptive approach, combining rule-based and behavior-based schemes, effectively distributed intrusion detection processing loads across wireless sensor nodes while minimizing energy consumption. In a similar vein, Riecker et al. [23] developed an energy-efficient system that is light weight, employing mobile agents for both intrusion detection and energy consumption monitoring. Their system incorporated linear regression modeling to predict energy consumption patterns across sensor nodes.

A notable advancement in this field is presented in [5], where researchers integrated deep learning with multi-agent systems to detect intrusions. This hybrid approach consolidated multi-agent system flexibility with deep learning precision through independent, intelligent, and adaptive agents implemented across three algorithms: k-nearest neighbors, autoencoder, and multilayer perceptron. The autoencoder serves as a feature reduction mechanism, while the multilayer perceptron along with the k-nearest neighbors function as classifiers. When tested on the KDD99 intrusion detection dataset, this hybrid distributed system achieved superior accuracy rates and significantly reduced detection times compared to traditional approaches.

Recent advancements in intelligent intrusion detection systems have demonstrated the effectiveness of deep learning approaches. Research by [24] introduced a Deep Q-Learning (DQ-L) model featuring continuous automatic learning capabilities designed for network environments. This model is capable of detecting different kinds of network intrusions based on an automated trial-and-error method, continually improving detection abilities. When assessed by the use of the NSL-KDD dataset, this model attained over 90% accuracy across different network intrusion classifications.

Building on reinforcement learning applications, [25] developed a NIDS using Deep Reinforcement Learning (DRL), specifically tailored for IoT environments and Wireless Sensor Networks. This system leverages Markov decision process formalism to enhance detection decision-making, incorporating K-Nearest Neighbors (KNN) for model construction. Comparative analysis demonstrated exemplar performance metrics, including improved detection rates and accuracy while reducing false alarms.

A significant contribution to distributed intrusion detection is presented in [26], where the researchers designed a reinforcement learning-based NIDS that utilized Deep Q-Network logic among several distributed agents, incorporating an attention approach. This approach effectively detects and classifies advanced network attacks. Extensive experimental evaluation utilizing CICIDS2017 and the NSL-KDD benchmark datasets validated the proposed model's superior performance, demonstrating dominance over over existing existing state-of-the-art systems through enhanced accuracy, recall, precision, F1-Score, and substantially reduced false-positive rates (FPR).

5. Proposed Network Intrusion Detection Model

5.1 Multi-Agent System Learning Model

The objective was to design a multi agent system model for detecting network intrusions leveraging machine learning capabilities. Meyer et al. [27] emphasize that developing intelligent agents through machine learning techniques, particularly reinforcement learning, demonstrates significant potential for future applications. The principal focus of reinforcement learning is on the optimization process that is motivated by random exploration and increasing exploitation of knowledge gathered from experiences.

The model operates under the assumption that agents function within an environment characterized by a Markov Decision Process, where decisions occur in discrete, stochastic, sequential environments [28]. The model comprises of a tuple (S, A, P, R, γ) where the set S contains all states that are possible and the set A consists all actions possible that the agent may choose. For every state $s \in S$ and a valid selected action $a \in A$ is succeeded by a transition to a subsequent state $s' \in S$. While these transitions can be deterministic in theory, practical implementations often encounter uncertainty factors, introducing probabilistic elements to the transition process.

Consequently, $P(s, a, s')$ defines the probability of a shift to the next state. Further, any shift from state s to the subsequent state s' through performing action a may give an outcome in an immediate value given by the reward function R, which may be sparse in some cases, meaning that not every new state necessarily has an instant value. P is the model determining transition probabilities that can be expressed in mathematical terms below:

$$P\left[S_{t+1} \middle| S_t\right] = P\left[S_{t+1} \middle| S_1, S_2 S_{3\ldots,} S_t\right] \qquad (1)$$

Figure 2 illustrates the reinforcement learning framework that is used in Markov Decision Processes.

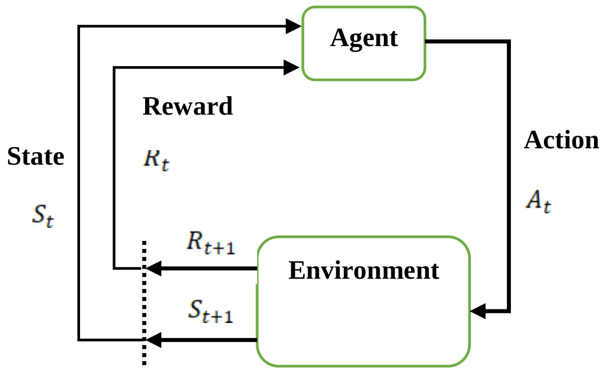

Figure 2. Reinforcement learning framework used in Markov Decision Processes (source: *[31]*)

The selection of reinforcement learning was driven by its inherent capability to develop effective policies within stochastic and noisy environments, provided appropriate hyperparameters are established. Our approach implements hierarchical reinforcement learning, which decomposes complex tasks into more manageable subtasks that can be learned independently and subsequently integrated into a comprehensive solution.

This hierarchical approach effectively operates as a Divide-and-Conquer strategy within the reinforcement learning framework, offering intuitive problem decomposition while demonstrating significant success in handling complex tasks [27], [30], [31]. The methodology has proven particularly effective in scenarios requiring sophisticated decision-making processes, as it enables systematic learning at multiple levels of abstraction while maintaining the ability to integrate learned behaviour into cohesive, higher-level strategies.

A model-free, value-based Q-Learning algorithm was used to optimize the model's state configuration. This algorithm operates through an iterative approximation process, initializing with an arbitrary Q-function and progressively refining it through observation and analysis of state transitions.

The method's core functionality lies in its ability to estimate the long-term projected reward when taking action, *a* from the given state, *s*. In successive iterations, the Q-function is continuously updated, enabling the model to learn optimal action-selection policies based on accumulated experience and calculated future rewards.

According to [32], Q-values, which represent the estimated long-term return of taking an action in a assumed state, are learned iteratively. The algorithm updates the present Q-value estimate by considering the immediate reward r_t and the greatest possible Q-value in all actions, a achievable in the subsequent state, s_{t+1}:

$$Q(s_t, a_t) \leftarrow (1-\alpha)Q(s_t, a_t) + \alpha\left(r_t + y.\max_a Q(s_{t+1}, a)\right) \tag{2}$$

Where $\gamma \in [0,1]$ represents the discounting factor and $a \in [0,1]$ as the learning rate.

Q-learning agents are designed to determine optimal policies that maximize total discounted rewards through environmental interaction [33]. The learning process occurs in discrete time steps, where agents perceive environmental states and execute actions, triggering state transitions. Each transition's quality is evaluated through a scalar reward signal, with agents striving to maximize cumulative rewards throughout their interaction sequence.

Multi-agent Q-learning encompasses two primary paradigms: best-response learning and equilibrium learning. The research implemented the best-response learning paradigm, where individual agents develop policies optimized relative to other agents' joint policies [34]. For complex systems with extensive state-action spaces, an alternative approach becomes necessary. Rather than maintaining individual state-action combination values, the system approximates state q-values using neural networks - function approximators composed of layered structures capable of pattern recognition in large-scale datasets. This approach enables efficient handling of high-dimensional state spaces while maintaining learning effectiveness.

A deep feed-forward network $f(x;\theta)$ learns to approximate a function $y = f * (x)$ by adjusting its parameters θ. Training a neural network corresponds to making adjustments to the parameters θ in such a way that the network learns to output values corresponding to a preferred result [35]. In reinforcement learning, neural networks serve as advanced mapping tools that convert state inputs into corresponding Q-values for the entire potential actions within that state. This representation enables the neural network to serve as a function approximator, effectively converting complex state information into actionable value estimates for decision-making.

5.2 Deep Q-networks

This study employed Deep Q-networks, pioneered by Google DeepMind, which integrates classical Q-learning principles with deep neural network architectures. This hybrid approach combines the value-based learning capabilities of Q-learning with the powerful function approximation abilities of deep neural networks to create a robust learning framework.

The Q network constructs a Q-table that comprises of State-Action values with dimensions |S| × |A|, in which |S| represents the state space cardinality whereas |A| denotes number of the possible actions. The network functions as a q-value approximator, where the policy network (Q network) processes state, *s* as an input and produces an array of q-values, producing one value to the individually available action, *a*. As such, the output of the neural network depends on its current weights θ, effectually making the output q-values a function of θ, $Q(s,a;\theta)$. The network Q is initialized and its weights θ set arbitrarily. Random q-values are returned when a state to the network is passed before it has undergone any training. However, after adequate training the network output converge towards the optimal q-value and the weights are updated by performing a gradient descent step with respect to θ. Figure 3 illustrates the workflow designed for the proposed Deep Q-network Model.

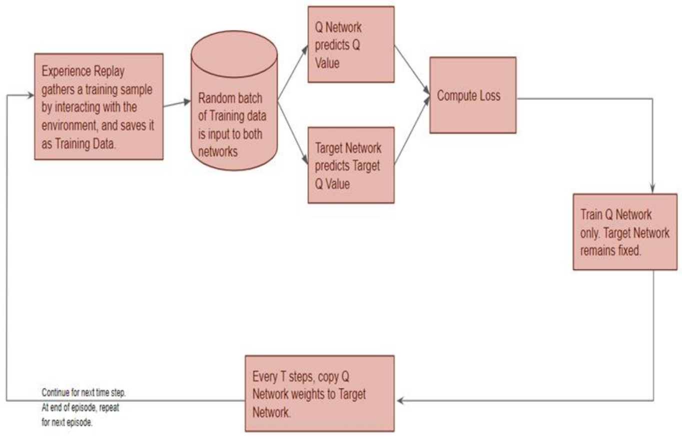
Experience Replay gathers a training sample by interacting with the environment, and saves it as Training Data.
Random batch of Training data is input to both networks
Q Network predicts Q Value
Target Network predicts Target Q Value
Compute Loss
Train Q Network only. Target Network remains fixed.
Every T steps, copy Q Network weights to Target Network.
Continue for next time step. At end of episode, repeat for next episode.

Figure 3. Workflow designed for the Deep Q Network Model (Source: *[36]*)

The proposed system architecture integrates three critical components: the Q network, the Experience Replay mechanism, and the Target network. The Q network serves as the primary intelligent agent, responsible for generating optimal state-action value estimations. Complementing this, the Experience Replay mechanism facilitates interaction with the environment, systematically capturing and storing transition data that enables efficient training of the Q network, thereby enhancing the overall learning process.

The Q network can be implemented through various neural network architectures: a linear network with multiple hidden layers for numeric state data, or Recurrent Neural Network (RNN) or a Convolutional Neural Network (CNN) for processing state data in the form of images or text. Experience Replay operates by selecting actions using an ε-greedy policy based on the existing state, executing these actions in the environment, and receiving corresponding rewards and subsequent states. These interactions are stored as training samples, as depicted in Figure 4.

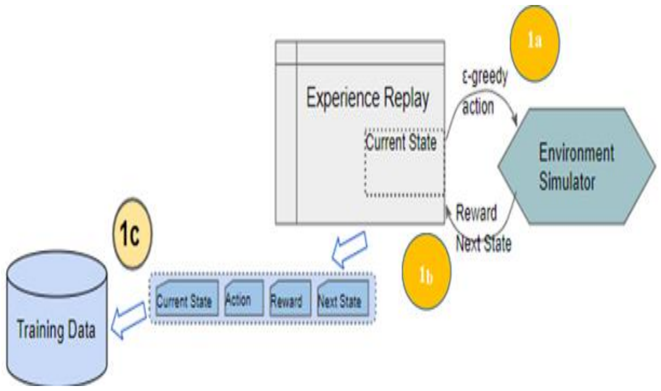

Figure 4. Illustrating the Experience Relay activity

The proposed deep Q-network employs two convolutional layers, a max-pooling layer, and three fully connected layers. Details of this architecture are provided in Table 1.

Table 1. Specifications for the Network architecture

Layer	Input	Filter Size	Stride	Activation	Output
Convolutional	$M \times M \times F_C$	2X2	1		$M \times M \times 32$
Convolutional	$M \times M \times 32$	2X2	1		$M \times M \times 32$
Max pooling	$M \times M \times 32$	2X2	2		$M/2 \times M/2 \times 32$
Fully Convolutional	$M/2 \times M/2 \times 32$			ReLu	512
Fully Convolutional	512			ReLu	256
Fully Convolutional	256			Softmax	8

5.3 Deep Q-Learning Model Architecture

This section examines the fundamental concepts of deep Q-Learning within the context of network intrusion detection, utilizing the CICIDS2017 benchmark dataset as the operational environment. This dataset provides the foundation for evaluating and implementing deep Q-Learning strategies in network security applications.

5.3.1 Environment

The environment is represented by the pre-processed and normalized CICIDS2017 dataset, where its features serve as states within the deep Q-Network architecture. Of the dataset's 81 features, 80 are utilized as state representations, while the 81st feature, the label, is employed for reward vector computation based on model predictions.

Significantly, in this proposed model, the agent's role is confined to action selection for reward vector computation, without executing actual modifications to the environment. This abstraction allows for effective model training while maintaining the integrity of the underlying dataset structure.

5.3.2 Agent

The proposed model employs deep agents structured according to the network architecture, with a minimum of one agent per network context. Agent-environment interactions generate rewards depending on current states as well as the selected actions. The deep Q-Network functions as a value-based reinforcement learning agent, trained to project anticipated reward outcomes through environmental interactions.

The learning procedure involves the agent's examination of the action space through the epsilon-greedy exploration policy [37]. This policy establishes a critical balance between exploration and exploitation by implementing a probabilistic decision mechanism: the agent employs an ε-greedy exploration strategy, where it arbitrarily pick out an action with probability ε, and alternatively selects the action with the maximum predictable value with the likelihood (1-ε), balancing exploration and exploitation of the action space. This procedure may be depicted in the following algorithmic blueprint:

```
P = random ()
    if  p < $\varepsilon$
        pull random action
    else:
        pull current-based action
```

5.3.3 States

In the deep Q-Network model, states represent the environmental inputs that guide agent actions. For this implementation using the CICIDS2017 dataset, the network's state parameters are derived from the dataset's features. Specifically, 80 features from the dataset serve as direct inputs to the deep Q-Network, creating a comprehensive state representation for decision-making processes.

5.3.4 Actions

An action represents the agent's decision output after processing environmental data within a specified time window, typically following mini-batch processing completion. The deep Q-Network agent generates an action vector that is dependent on the neural network's input features. Through multiple episode iterations, the Q-learning function maintains and updates running averages of Q-values until they converge to optimal values.

These optimized Q-values serve dual purposes: determining successful attack detection and feeding state vectors (sized according to mini-batch dimensions) into the current deep Q-Network. The agent then performs classification by comparing the deep Q-Network outputs against established threshold rates, utilizing Q-threshold values to categorize different attack classes.

5.3.5 Rewards

A reward constitutes environmental feedback generated in response to an agent's action, formulated as a reward vector based on deep Q-Network output values and mini-batch dimensions. Within the context of the CICIDS2017 dataset, reward allocation follows a binary principle: positive rewards are assigned when the deep Q-Network's classification aligns with actual dataset labels, while misclassifications result in negative rewards.

The magnitude of rewards can be calibrated according to the classifier's prediction probability, with further adjustments possible based on achieved Q-values to optimize classifier performance. This dynamic reward structure enables continuous model refinement through reinforcement of accurate predictions and penalization of errors.

6. Experiments

6.1 The CICIDS 2017 Dataset

Model evaluation utilized the CICIDS2017 dataset [3], selected for its comprehensive coverage of diverse network attacks. The dataset consists of network traffic pcap files collected over a five-day period (Monday through Friday), with labeled attack samples present in all days except Monday, providing an ideal structure for detection evaluation.

The experimental design designated Monday's traffic file for model training, while files from Tuesday through Friday were used to verify detection accuracy. The detailed composition of the CICIDS2017 dataset utilized in the study is presented in Table 2.

Table 2. CICIDS2017 Dataset

Traffic Type	Attack	Training Data	Testing Data
Benign		249,044	150,618
Brute Force	FTP-Patator	-	2,457
	SSH-Patator	-	2,905
DoS/DDOS	Slowloris	-	3,518
	Slowhttptest	-	3,610
	Hulk	-	9,535
	GoldenEye	-	6,592
Web Attack	Bruteforce	-	143
	XSS	-	18
	SQL-Injection	-	7
Bot		-	1,207
Port Scan		-	154,571

The dataset was divided into two segments to ensure effective model development and evaluation. The training set, comprising 70% of the total dataset, incorporates all categories of network events, including both benign and malicious traffic patterns. This comprehensive inclusion enables the model to learn from the full spectrum of possible network events. The remaining 30% was allocated as the test set, serving as independent data for evaluating the model's performance and validating its detection capabilities across various attack scenarios.

6.2 Experimental set up

Performance evaluation of the proposed model was conducted through a series of experiments. The model was implemented using Python programming language, leveraging TensorFlow backend along with Keras, NumPy, and scikit-Learn packages for comprehensive functionality. Table 3 outlines the experimental environment specifications.

Table 3. Implementation environment specifications

Unit	Description
Processor	Intel Core i5 CPU with 2.50 GHz
Memory	8GB
Operating System	Microsoft Windows 10 Pro.
Statistical and visualization Packages	Keras, Tensorflow, Numpy, Sci-kit-Learn, Pandas, and Matplotlib

The model's performance was evaluated using key performance metrics: Accuracy, Precision, True Positive Rate (TPR - also known as recall or detection rate), False Alarm Rate (FAR), and F1-measure. The metrics are derived from a confusion matrix presented in Table 4, using the resulting equations:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} * 100\% \qquad (3)$$

$$TPR(\mathrm{Re}\,callorDetectionRate) = \frac{TP}{TP + FN} * 100\% \qquad (4)$$

$$\mathrm{Pr}\,ecision = \frac{TP}{TP + FP} * 100\% \qquad (5)$$

$$F1 - Measure = \frac{2 * precision * \mathrm{Re}\,call}{precision + \mathrm{Re}\,call} * 100\% \qquad (6)$$

Performance evaluation in network security detection relied on key statistical measures derived from the confusion matrix. This matrix breaks down classification results into four main categories: True Positives (TP) capture successfully detected attacks, True Negatives (TN) confirm correctly recognized regular network traffic, False Positives

(FP) represent benign traffic erroneously labelled as malicious, and False Negatives (FN) signify attacks that have been overlooked and misclassified as legitimate network activity.

Table 4. The Confusion Matrix

		Predicted Value	
Actual Value		**Positive**	**Negative**
	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

7. Results and Discussion

The model validation was conducted through a rigorous 10-fold cross-validation process, employing multiple performance parameters including Accuracy, Recall, Precision, and F-1 score. Accuracy serves as a fundamental metric, quantifying the model's prediction perfection by comparing its outputs against true data values. This thorough validation methodology rigorously assesses the model's predictive performance by examining its accuracy and reliability across multiple data segments.

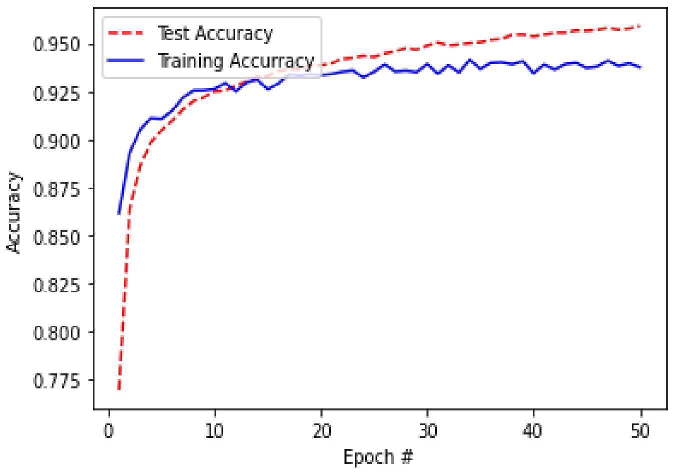

Figure 5. Model accuracy result

The model exhibited excellent performance in attack detection, achieving a 97.71% detection accuracy rate after completing 50 iterations, as depicted in Figure 5. These notable results demonstrate the model's effectiveness in identifying and classifying known attack patterns within the CICIDS2017 dataset, demonstrating its robust capability in network intrusion detection.

The 97.71% accuracy is particularly noteworthy, as it represents a substantial improvement over traditional NIDS. This high accuracy rate indicates the multi-agent deep reinforcement learning (MADRL) approach's exceptional ability to distinguish between normal network traffic and various attack patterns. Further, the achievement of this metric after only 50 iterations is particularly impressive, highlighting the efficiency of the proposed multi agent deep reinforcement learning approach. This learning strategy appears to be crucial in this success, effectively decomposing complex intrusion detection tasks into more manageable subtasks.

The research's contribution extends beyond mere this technical achievement. It represents a critical step in developing more intelligent, adaptive network security systems that can keep pace with the rapidly evolving landscape of cyber threats. The proposed framework's ability to effectively learn and mimic environmental characteristics through multi-agent learning provides a robust foundation for future advancements in intrusion detection technologies.

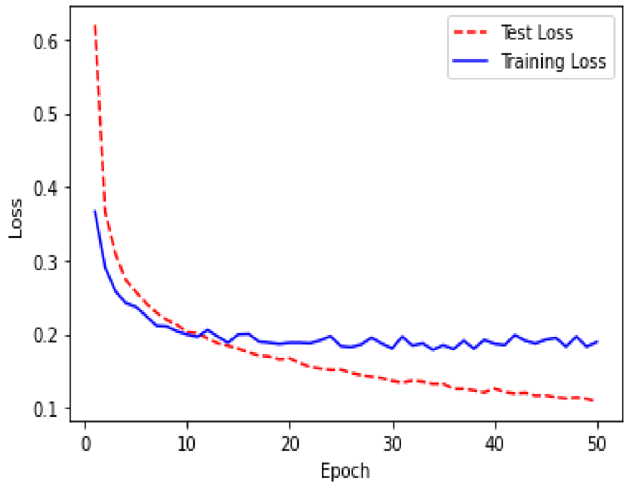

Figure 6. Model loss result

The model further demonstrated excellent performance, evidenced by the minimal divergence between training and testing loss values, as shown in Figure 6. Performance visualization through the confusion matrix illustrated the relationship between true positives and false negatives. The comprehensive evaluation metrics, including Accuracy, Precision, True Positive Rate, and the F1-score), are presented in Table 5, quantifying the model's robust classification capabilities.

Table 5. Performance metrics of the model trained on loss functions after 50 epochs

Loss Function	Accuracy (%)	Validation Error	Recall	Precision	F1 score
Categorical cross-entropy	97.71	0.0254	0.9834	0.9729	0.9781

The model demonstrated robust performance across key metrics: achieving 98.34% sensitivity (recall) in correctly identifying positive class observations, and 97.29% precision in positive predictions accuracy. The harmonic mean of these measures, represented by the F1-score, reached 97.81%. The model's generalization capability was validated using an independent test set, revealing strong performance on unseen data and confirming its effective learning of intrusion patterns rather than mere memorization.

A comparative analysis of the proposed model against recent solutions was conducted, with benchmarking results presented in Table 6, positioning our model's performance within the current research landscape.

Table 6. Type Styles (Size 8 Normal)

Ref.	Dataset	Accuracy	Recall	Precision	F-1 score
[39]	NSL-KDD	80.00%	-	-	79.00 %
[24]	NSL-KDD	78.07%	76.76 %	77.84%	81.41 %
[5]	KDD 99	99.95%	99.88 %	99.88%	99.88 %
Proposed Model	CICIDS 2017	97.71%	98.34 %	97.29%	97.81 %

The performance metrics presented above reveal the model's exceptional capability in network intrusion detection. The high sensitivity (recall) of 98.34% is particularly significant, as it indicates the model's near-comprehensive ability to detect actual attack instances. This metric is critical in network security, where missing a potential threat can have severe consequences.

The precision of 97.29% complements the high recall, demonstrating the model's remarkable capacity to reduce the false positives. In the context of network security, this balance is crucial. High precision ensures that legitimate network traffic is not unnecessarily disrupted, while the high recall guarantees that potential threats are not overlooked.

The F1-score computed as 97.81% gives a holistic view of the performance of the proposed model, representing an optimal sense of balance between precision and recall. This metric is particularly valuable in imbalanced datasets typical of network intrusion detection, where attack instances are usually rare compared to normal network traffic.

The model's generalization capability, validated through an independent test set, is a critical validation of its learning approach. By performing strongly on unseen data, the research demonstrates that the proposed model has learned meaningful intrusion patterns rather than simply memorizing the training dataset. This characteristic is essential for developing adaptive and robust network security solutions that can respond to evolving cyber threats.

The comparative analysis against recent solutions (as referenced in Table 6) is key for contextualizing the model's performance within the current research landscape. Such benchmarking not only highlights the model's advances but also provides a clear perspective on its contributions towards the domain of network intrusion detection.

The research's approach of using a multi agent deep reinforcement learning approach appears to offer significant advantages over traditional methods. By decomposing complex detection tasks and employing advanced learning strategies, the proposed model demonstrates a sophisticated approach to addressing the increasingly sophisticated landscape of network security challenges.

The high performance across multiple metrics suggests that the resultant model could potentially bridge existing gaps in NIDSs. Its ability to accurately recognize and classify attack patterns while maintaining low false positive rates makes it a promising solution for real-world network security implementations.

Moreover, the model's success underscores the potential of advanced machine learning techniques, particularly deep reinforcement learning, in addressing complex cybersecurity challenges. The research, despite providing a novel technical solution, also offers a methodological framework that could inspire future innovations in the domain of network security and threat detection.

8. Conclusion and Future Scope

This research significantly advances the domain of Network Intrusion Detection Systems (NIDS) by introducing an innovative multi agent deep reinforcement learning approach. The developed attack prediction model, employing a deep Q-network (DQN) architecture with convolutional and fully connected layers, demonstrates remarkable effectiveness in identifying and mitigating complex cyberattacks. The integration of Target networks and Experience Replay enhances the learning process, enabling the model to adapt to evolving threat landscapes.

The study's findings, based on rigorous evaluation using the CICIDS2017 dataset, highlight the model's exceptional performance. Achieving 97.71% accuracy, 98.34% recall, 97.29% precision, and 96.76% F1-score, the model surpasses existing NIDS solutions, particularly in handling high-dimensional state-action spaces and intricate attack patterns. This success emphasizes the likelihood of multi agent deep reinforcement learning in improving network security.

The model's ability to effectively mimic environmental characteristics through multi-agent learning opens up new avenues for addressing the challenges posed by sophisticated cyber threats. By decomposing complex intrusion detection tasks into manageable subtasks, the hierarchical reinforcement learning strategy facilitates efficient learning and adaptation. This approach enables the model to effectively explore the vast state-action space, leading to improved detection accuracy and robustness.

Furthermore, the model's strong generalization capabilities on unseen data demonstrate its potential for real-world deployment. This adaptability is crucial in ensuring the long-term effectiveness of NIDS, as cyberattacks continuously evolve and become more sophisticated.

Building upon the promising results of this research, several avenues for future exploration emerge:

i. **Real-time Implementation:** Deploying the model in real-time network environments to assess its performance and scalability under dynamic conditions is crucial. This involves integrating the model with existing network infrastructure and evaluating its efficiency in processing high-volume network traffic.

ii. **Agent Network Expansion:** Expanding the agent network and exploring different communication and coordination mechanisms among agents could further enhance the model's detection accuracy. This could involve investigating hierarchical structures, decentralized approaches, or the use of attention mechanisms to improve collaboration.

iii. **Outlier Detection:** Extending the model's application to outlier detection can enable the identification of anomalous network behaviour that may indicate novel or previously unseen attack patterns. This can contribute to proactive threat mitigation and enhance the overall security posture.

iv. **Software-Defined Networking (SDN):** Integrating the model with SDN frameworks can enable dynamic and automated responses to detected threats. This allows for flexible network reconfiguration and security policy enforcement, facilitating adaptive security measures.

By pursuing these research directions, this study provides a good foundation for developing more sophisticated and robust NIDS. The use of multi agent deep reinforcement learning holds immense promise for revolutionizing network security and safeguarding critical infrastructure in spite of the evolving cyber threats.

Data Availability
None.

Conflict of Interest
The authors hereby declare that they have no financial or non-financial conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding Source
None.

Authors' Contributions
All authors contributed to the manuscript's revision, editing, and approved its final version for publication.

Acknowledgements
None.

References

[1] H. Liao, C. Lin, Y. Lin and K. Tung, "Intrusion detection system: a comprehensive review," *Journal of Network Computing Applications*, Vol. **36**, No. **1**, pp. **16-24**, **2013.**

[2] K. Harmanpreet and K. Harjot, "Using Multi-Agent Systems for Intrusion Detection in Computer Networks: A Glance," *International Journal of Advanced Research in Computer Science*, Vol. **9**, No. **2**, pp. **497-500**, **2018**.

[3] I. Sharafaldin, A. Gharib, A. H. Lashkari and A. Ghorbani, "Towards a Reliable Intrusion Detection Benchmark Dataset," *Journal of Software Networking*, Vol. **2017**, No. **1**, pp. **177–200**, **2017**.

[4] Y. Wu, D. Wei and J. Feng, "Network Attacks Detection Methods Based on Deep Learning: A Survey," *Security and Communication Networks*, Vol. **2020**, No. **Article ID 8872923**, pp. **17**, **2020**.

[5] F. Louati and F. Ktata, "A deep learning-based multi-agent system for intrusion detection," *SN Applied Sciences*, Vol. **2**, No. **675**, pp. **1-13**, **2020.**

[6] V. Jyothsna and K. Prasad, "Anomaly-Based Intrusion Detection System," in *Computer and Network Security*, IntechOpen, pp. **1-15**, **2019**.

[7] S. A. Althubiti, E. M. Jones and K. Roy, "LSTM for Anomaly-Based Network Intrusion Detection," *in 2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, Sydney, NSW, **2018**.

[8] M. Wooldridge and N. R. Jennings, "Intelligent agents: theory and practice," *The Knowledge Engineering Review*, Vol. **10**, No. **2**, pp. **115-152**, **1995**.

[9] J. Fox, M. Beveridge and D. Glasspool, "Understanding intelligent agents: analysis and synthesis," *AI Communications*, Vol. **16**, No. **3**, pp. **139-152**, **2003**.

[10] K. Monu and W. Carson, "Intelligent Agents as a Modeling Paradigm," *in ICIS 2005 Proceedings*, Las Vegas, NV, USA, **2005**.

[11] G. Weiss, "Agent Orientation in Software Engineering," *Knowledge Engineering Review*, Vol. **16**, No. **4**, pp. **349-373**, **2002**.

[12] A. Hrebennyk and E. Trunova, "Modelling a Multi-agent Protection System of an Enterprise Network," *ISIJ Monitor*, Vol. **46**, No. **3**, pp. **337-340**, **2020**.

[13] F. Derakhshan and S. Yousefi, "A review on the applications of multiagent systems in wireless sensor networks," *International Journal of Distributed Sensor Networks*, Vol. **15**, No. **5**, pp. **1-19**, **2019**.

[14] Á. Herrero and E. Corchado, "Multiagent Systems for Network Intrusion Detection: A Review," *Computational Intelligence in Security for Information Systems. Advances in Intelligent and Soft Computing*, Springer, Berlin, Heidelberg, pp. **143-154**, **2009**.

[15] T. T. Nguyen, N. D. Nguyen and S. Nahavandi, "Deep Reinforcement Learning for Multi-Agent Systems: A Review of Challenges, Solutions and Applications," *IEEE Transactions on Cybernetics*, Vol. **50**, No. **9**, pp. **3826-3839**, **2020.**

[16] Y. Shoham, R. Powers and T. Grenager, "Multi-agent reinforcement learning: a critical survey. Tech. rep.," Stanford University, **2003**.

[17] R. Lowe, Y. Wu, A. Tamar, J. Harb, O. Abbeel and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments.," *in Advances in neural information processing systems*, pp. **6382-639**, **2017**.

[18] Y. Miyashita and T. Sugawara, "Analysis of coordinated behavior structures with multi-agent deep reinforcement learning," *Applied Intelligence*, Vol. **51**, pp. **1069-1085**, **2021**.

[19] N. Liu, S. Liu, R. Li and Y. Liu, "A Network Intrusion Detection Model Based on Immune Multi-Agent," *International Journal of Communications, Network and System Sciences*, Vol. **2**, No. **6**, pp. **569-574**, **2009**.

[20] S. Ouiazzane, F. Barramou and M. Addou, "Towards a Multi-Agent based Network Intrusion Detection System for a Fleet of Drones," *International Journal of Advanced Computer Science and Applications (IJACSA)*, Vol. **11**, No. **10**, pp. **351-362**, **2020**.

[21] Y. Zhang and W. Lee, "Intrusion detection in wireless ad-hoc networks," *in Proceedings of the 6th annual international conference on mobile computing and networking*, MobiCom, Boston, MA, **2000**.

[22] D. Krishnan, "A Distributed Self-Adaptive Intrusion Detection System for Mobile Ad-hoc Networks Using Tamper Evident Mobile Agents," *Procedia Computer Science*, Vol. **46**, pp. **1203-1208**, **2015**.

[23] M. Riecker, S. Biedermann, R. El Bansarkhani and M. Hollick, "Lightweight energy consumption-based intrusion detection system for wireless sensor networks," *International Journal of Information Security*, Vol. **14**, pp. **155-167**, **2015**.

[24] H. Alavizadeh, H. Alavizadeh and J. Jang-Jaccard, "Deep Q-Learning Based Reinforcement Learning Approach for Network Intrusion Detection," *Computers*, Vol. **11**, No. **41**, pp. **1-19**, **2022**.

[25] H. Benaddi, K. Ibrahimi, A. Benslimane and J. Qadir, "A Deep Reinforcement Learning Based Intrusion Detection System (DRL-IDS) for Securing Wireless Sensor Networks and Internet of Things," *in Wireless Internet. WiCON 2019. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Vol. **317**, Springer, Cham., **2020**.

[26] K. Sethi, Y. V. Madhav, R. Kumar and P. Bera, "Attention based multi-agent intrusion detection systems using reinforcement learning," *Journal of Information Security and Applications*, Vol. **61**, **2021**.

[27] O. Meyer, M. Hesenius and V. Gruhn, "Using Concepts to Understand Intelligent Agents," *Proceedings of the AAAI 2020 Spring Symposium on Combining Machine Learning and Knowledge Engineering in Practice (AAAI-MAKE 2020)*, pp. **9**, **2020**.

[28] E. A. Feinberg and A. Shwartz, “Handbook of Markov Decision Processes: Methods and Applications,” Springer Science & Business Media, **2012**.

[29] X. Li, H. Zhong and M. L. Brandeau, "Quantile Markov Decision Processes," *Operations Research*, Vol. **70**, No. **3**, **2021**.

[30] A. Gudimella, R. Story, M. Shaker, R. Kong, M. Brown, V. Shnayder and M. Campos, "Deep Reinforcement Learning for Dexterous Manipulation with ConceptNetworks," *arXiv preprint*, p. arXiv:1709.06977, **2017**.

[31] T. D. Kulkarni, K. Narasimhan, A. Saeedi and J. Tenenbaum, "Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation," *in Advances in neural information processing systems*, pp. **3675-3683**, **2016**.

[32] M. Zhou, Y. Chen, Y. Wen, Y. Yang, Y. Su, W. Zhang, D. Zhang and J. Wang, "Factorized Q-Learning for Large-Scale Multi-Agent Systems," *arXiv Preprints*, p. arXiv:1809.03738v4, **2019**.

[33] L. Buşoniu, R. Babuˇska and B. De Schutter, "Multi-agent reinforcement learning: An overview," *in Innovations in Multi-Agent Systems and Applications*, Berlin, Germany, Springer, pp. **183-221**, **2010**.

[34] C. Claus and C. Boutilier, "The dynamics of reinforcement learning in cooperative multiagent systems," *in Proceedings of the 15th National Conference on Artificial Intelligence (AAAI) and 10th Innovative Applications of Artificial Intelligence Conference*, Madison, WI, US, **1998**.

[35] I. Goodfellow, Y. Bengio and A. Courville, “Deep Learning”, MIT Press, **2016**.

[36] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra and M. Riedmiller, "Playing Atari with Deep Reinforcement Learning," *arXiv Preprints*, p. arXiv:1312.5602, **2013**.

[37] M. Tokic, "Adaptive ε-greedy exploration in reinforcement learning based on value differences," *in Advances in Artificial Intelligence*, Berlin / Heidelberg, Springer, pp. **203-210**, **2010**.

[38] E. Suwannalai and C. Polprasert, "Network Intrusion Detection Systems Using Adversarial Reinforcement Learning with Deep Q-network," *in 2020 18th International Conference on ICT and Knowledge Engineering (ICT&KE)*, Bangkok, Thailand, **2020**.

[39] N. Singh, S. Krishan and U. K. Singh, "An Enhanced Multi-Agent based Network Intrusion Detection System using Shadow Log," *International Journal of Computer Applications*, Vol. **100**, No. **9**, pp. **1-5**, **2014**.

AUTHORS PROFILE

Dr. Stephen Kahara is a distinguished academic and technology professional currently serving as a Lecturer of Computer Science at Murang'a University of Technology and Director of Performance Contracting and ISO. With a comprehensive academic background including a Ph.D. in Computer Science from Murang'a University of Technology (2023), an M.Sc. in Computer Systems from JKUAT (2018), an M.Sc. in Organizational Development from USIU-Africa (2010), and a B.Sc. in Information Sciences from Moi University (2006), Dr. Kahara brings profound interdisciplinary expertise to his research and professional roles. Possessing over 16 years of experience in the ICT industry, he is a certified QMS and ISMS auditor and DAAD-UNILEAD alumnus. His research interests strategically focus on machine learning, network security, distributed systems, and computational biology. Dr. Kahara has established a robust scholarly reputation through extensive publications in reputable international journals and conferences, consistently contributing to advancing technological knowledge. As a dedicated member of the Association of Computing Practitioners - Kenya, he remains actively engaged in professional development and technological innovation.

Dr. Stephen Njenga earned his Ph.D. in Information Systems and M.Sc. in Computer Science from the University of Nairobi, and B.Sc. in Computer Science from Egerton University. He currently serves as a Lecturer in the Department of Computer Science at Murang'a University of Technology. With 15 years in academia and 8 years of research experience, he has published extensively in reputable international journals and IEEE conferences. His research interests center on machine learning, distributed ledger technology, and intelligent agents.