TRUST METRIC-BASED ANOMALY DETECTION VIA DEEP DETERMINISTIC POLICY GRADIENT REINFORCEMENT LEARNING FRAMEWORK

Shruthi N[1] and Siddesh G K[2]

[1]Research Scholar, JSS Academy of Technical Education,
Visvesvaraya Technological University, Belagavi-590018, Karnataka
[2]Head, ECE Department, ALVA's Institute of Engineering & Technology,
Moodbidri-574225, Karnataka

ABSTRACT

Addressing real-time network security issues is paramount due to the rapidly expanding IoT jargon. The erratic rise in usage of inadequately secured IoT- based sensory devices like wearables of mobile users, autonomous vehicles, smartphones and appliances by a larger user community is fuelling the need for a trustable, super-performant security framework. An efficient anomaly detection system would aim to address the anomaly detection problem by devising a competent attack detection model. This paper delves into the Deep Deterministic Policy Gradient (DDPG) approach, a promising Reinforcement Learning platform to combat noisy sensor samples which are instigated by alarming network attacks. The authors propose an enhanced DDPG approach based on trust metrics and belief networks, referred to as Deep Deterministic Policy Gradient Belief Network (DDPG-BN). This deep-learning-based approach is projected as an algorithm to provide "Deep-Defense" to the plethora of network attacks. Confidence interval is chosen as the trust metric to decide on the termination of sensor sample collection. Once an enlisted attack is detected, the collection of samples from the particular sensor will automatically cease. The evaluations and results of the experiments highlight a better detection accuracy of 98.37% compared to its counterpart conventional DDPG implementation of 97.46%. The paper also covers the work based on a contemporary Deep Reinforcement Learning (DRL) algorithm, the Actor Critic (AC). The proposed deep learning binary classification model is validated using the NSL-KDD dataset and the performance is compared to a few deep learning implementations as well.

KEYWORDS

Deep Deterministic Policy Gradient, Reinforcement Learning, Security, Anomaly Detection, Confidence Interval, LSTM, Actor Critic.

1. INTRODUCTION

Anomaly detection is one such viable area of research dealing with real-time detection of cyber-attacks and threats. Anomalies can either bebased on the type of data (behavioural) or amount of data (volume) and can reflect as one of the following - an abnormality in data, unusual data patterns or faulty data packets, absurd increase in data packets, unusual unexpected behaviour of the network or change of distribution of packets at ports and speed variations.

Anomaly detection-related contributions would be of great help in counterattacking powerful network attacks. Logical security measures - authentication, authorization, encryption mechanisms, protocols and algorithms must be made available at the core cloud, edge servers, edge networks and the edge devices [1]. The work referred to in this paper focuses on catering to

DOI: 10.5121/ijcnc.2023.15601

security at edge devices, which are intelligent nodes equipped with data-gathering sensors. Data samples received from sensors will be tested for malicious activity and undesired anomalies. Machine Learning (ML) tools techniques and algorithms are widely used in different

domains and are capable of detecting network anomalies automatically [2]. Deep Learning (DL)is a well-suited fit to handle large-scale network traffic belonging to larger datasets. The related work in [3] states that the best deep learning models reduce the error rate by a considerably good percentage when compared to shallow machine learning models. DL is known for distributed computing and analysis of unlabelled and uncategorized data [4]. Reinforcement learning (RL) is an imposing type of DL technique which secures data transfer efficiently at the network edges. RL is fundamentally based on a “reward” function and the agent learns from the critic feedback post-environmental interaction. This concept of “dynamic feedback-oriented learning” is well suited for edge environments which handle real-time sensitive data.

1.1. Contributions

The contributions of the proposed work aim to propose the following:

1. A binary indicative, robust adversarial attack detection model based on the posterior trust-based value in reward calculation.
2. DDPG framework-based implementation for improved detection accuracy.
3. Long Short-Term Memory (LSTM) network architecture-based model for temporal dynamics of the edge sensors.

1.2. Organization of the Paper

Section 2 provides an overview of different edge attacks and proposed countermeasures followed by the role played by DL in Edge security. The section also opens up about the single- tailed function for anomaly detection using a Null Hypothesis based on Confidence Intervals. Section 3 details correspond to the results of using Supervised and Unsupervised Learning algorithms on the selected dataset. Section 4 discusses how the DL algorithms are classified. The final subsection here throws more light on the DDPG framework which is the core framework for the proposed work. The System model design and equations, problem formulation and the Network Architecture for DDPG based on LSTM networks are part of Section 5. Section 6 reveals the underlying algorithm for implementation. Section 7 encloses all the related results which justify the authors’ work. Section 8 gives an outline of the concluding notes along with a proposal for the future.

2. RELATED LITERATURE

2.1. Edge Attacks and Countermeasures

The authors of [5] mention in their work a set of attacks which supposedly constitute edge computing attacks. The four main attack categories are discussed briefly.

Distributed Denial of Service (DDoS) attacks are caused when the attacker sends an uncontrollable stream of data packets to the victim thereby draining its resources. In such situations, legitimate requests cannot be handled by the victim. *Flooding-based DDoS attacks* are practically prevalent in edge computing systems since most of the edge devices possess limited computational power and are easily targeted by attackers. One such attack was the Mirai [6] where compromised devices morphed as bots launched attacks on the edge servers, severely impacting the network. The earlier proposals for Per-packet-based detection of flooding-based attacks identified the DDoS packets based on packet identifiers [7] and checked for legitimate IP addresses of the DDoS packets [8]. On the contrary, Statistics-based approaches did not either require monitoring per packet information or have a repository of IP addresses, unlike the former, which used packet entropy and/or machine learning tools. The authors propose the D-WARD defence system in [9]. Authors of [10] use monitored source IP addresses, Hidden Markovian models and RL in their solution. Solutions based on Support Vector Machines (SVMs) and Genetic algorithms (GA) also project themselves as a viable solution to DDoS detection [11]. Zero-Day attacks, another headstrong, advanced group of DDoS attacks can result in memory corruption and service shutdowns. The authors put forth a memory isolation extension module to defend against possible memory corruption attacks [12]. Other solutions include software-defined networking (SDN) based IoT firewall to reduce the attack surface of an exposed IoT device [13], and lightweight isolation mechanisms on access routers to mitigate the damage of edge devices [14]. A noticeable approach is mentioned in [15] where the work focuses on reducing False Positive Rate (FPR). The authors of [16] provide a deep learning- based “Deep-Defense” approach which is based on Recurrent Neural Networks (RNNs). Another noticeable work is the use of dynamic threshold value in a statistical approach to formulate a DDoS detection model [17].

Malware injection attacks are both server-sided and device-sided. ML-based solutions for SQL and XSS detection were discussed in [18] and [19] respectively.

Side Channel attacks use publicly accessible information/ side channel information which is correlated with the privacy-sensitive data by the attacker. Solutions include data perturbation technique (differential privacy), a differentially private platform for data computation over the edge servers [20] and source code level discombobulation.

Authentication and authorization attacks are executed by the attackers via unauthorized access. Possible defence mechanisms against authentication attacks have to ensure the security of the communication protocols used in edge computing (WPA/WPA2, OAuth and SSL/TLS).

TABLE I: Overview of Network Attacks in Edge environments

Edge Attack	**Categories**	**Examples**	**Countermeasures**
DDoS attacks	a)Application layer b)Volumetric Protocol	GET/POST, Low-and-Slow POST, Single session/request, Fragmented HTTP flood, Recursive GET flood, Random Recursive GET flood UDP flood, CharGEN flood, ICMP flood, ICMP Fragmentation flood IP Null, TCP Flood, Session, Slowloris, Ping of Death, Smurf, Fraggle, Low Orbit Ion Cannon, High Orbit Ion Cannon	Hidden Markov models, ML-based Defense mechanisms.
Malware Injection attacks	a)Server side b)Device side	SQL injection, LDAP injection, Email injection, CRLF injection, Code injection, Cross-site scripting, OS Command injection, Host Header injection, XPath injection, wrapping attack, False Data Injection attacks	Signature-based detection, Blocklisting file extension, malware honeypot, cyclic redundancy checks, entropy-based dynamic analysis.

Side Channel attacks	a)Power consumption b)Electromagnetic c)Timing d)Fault Analysis	Wave signals, Data packets from sensors, acoustic, shared CPU caches, leakage from cryptographic devices	Differential privacy techniques
Authenticat ion & Authorization attacks	a)Insufficient Authentication b)Weak Password Recovery	Spear Phishing, Broad-based Phishing, Credential stuffing, Password Spraying, Brute Force attack, Man-in-the-middle attacks	Active jammers, Black box verification, public key cryptography, wireless packet injection, cross-layer authentication

2.2. Role of Deep Learning in EC Security

A Deep Neural network has several layers wherein each layer processes the intermediate characteristics of the previous layer and generates new characteristics [21]. Edge computing is efficient for deep learning tasks since the size of the extracted features is reduced by the filters in deep network layers. A detailed review of DL in Edge Computing (EC) security is provided by the authors in their work [22]. The related work in our paper focuses on security at the edge devices and therefore the discussion needs to touch upon the main reason for choosing DL in edge computing. Edge computing offloads computing tasks from the centralized cloud to the edge of IoT devices and pre-processing reduces the transferred data. The multi-layered, deep learning model helps in low-dimensioning or reducing data size, progressing over the network layers. Edge processing eases if the intermediate data size is smaller than the input data. Therefore, one can affirm that deep learning modelsare suitable for the edge computing environment wherein sections of the learning layers can be offloaded in the edge and the reduced intermediate data can then be transferred to the centralized cloud server [23]. The automated feature learning characteristic of the deep-learning-based models and choice of appropriate datasets significantly increases the detection rate accuracy compared to the preliminary ML algorithms [24].

We propose an effective Reinforcement Learning (RL) based security approach for edge security in comparison with the Supervised Learning (SL) and Unsupervised Learning (USL) counterparts. Q-learning enforced high-ambit issues in edge security solutions. The authors in [25] discuss an on-policy, Actor-Critic-based algorithm for anomaly detection in edge environments.

2.3. Confidence Interval-based Anomaly Detection Systems

Since the sensor samples are from a stochastic environment, it is suggested to coin the posterior trust metric (Ψ) with a probability of an anomalous detection (ρ) which varies proportionally with (Ψ). It is also important to note that (Ψ) affects the confidence interval as well.

A reported confidence interval is a range between two numbers within which the probability of containing the right value of a parameter exists. The typical value of 95% refers only to how often 95% confidence intervals computed from very many studies would contain the true size if all the assumptions used to compute the intervals were correct [26]. The remaining 5% constitutes the level of significance which is discussed in the next subsection.

2.4. Null Hypothesis using Single-Tailed Function for Anomaly Detection

Herman Chernoff proposed the active hypothesis test in 1959 [27]. The related work considers the processes (P_r) as the samples obtained from sensors. Sensor data is used to assess the believability and validity of a hypothesis. This is what is referred to as ”Hypothesis testing.” The objective is to architecture a model which stands by or rejects the framed hypothesis for a set of observations/samples $\{O_1, O_2, ...O_{P_r} \in (0,1)\}$ from sample space S(t), samples being captured from a particular sensor at varying time instants (t_1, t_2, …..t_z). The model has to then learn and master the optimal selection policy.

The hypothesis testing problem equivalent to the anomaly detection problem has a $2P_r$ hypothesis. The null hypothesis is a condition of the system that is not required i.e. system has encountered a network attack, it is a negation of the research question. As long as the null hypothesis test (H_i : i = 1,2,….2 P_r) is “false”, samples will be collected from the sensor else the supply chain has to be terminated. Real-time scenarios are such that the number of anomalous processes (P_r) is definitely lesser than the total number of processes (say K) i.e. (P_r << K) thereby conceptualizing that anomalous processes are rare events in a larger scenario of processes. Poisson distribution models rare events, thereby motivating the researcher to go ahead with an asymmetric distribution skewed to the right, inhibited by the zero-occurrence barrier to the left and extending towards the right. Poisson distribution can be represented as below:

$$P(X = x) = \frac{\lambda^x e^{\lambda}}{x\,!} \tag{1}$$

- P(x) = Probability of x successes given an idea of λ
- λ= Average number of successes
- e = 2.71828
- x = successes per unit which can take values 0,1,2,3,... ∞

The statistical hypothesis tests to accept or reject the null hypothesis are formulated using tailed functions. We use One-tailed tests for asymmetric distributions that have a single tail. The tail in the hypothesis test refers to the tail end at either side of the distribution curve. The Level of significance ($\boldsymbol{\alpha}$) needs to be fixed before the hypothesis since it conveys how wrong we are permitting the hypothesis to be, it is the probability of making wrong decisions when the null hypothesis is true. $\boldsymbol{\alpha}$ value is typically around 5%, as proposed by Fisher. However, this approach can be misleading for larger data samples, resulting in too frequent rejections of the null hypothesis. The level of significance depends on sample size, power of test, and expected losses from Type I and Type II errors. Also, an able mathematician Irving J. Good proposed a method for scaling the p-value cut-off according to sample size in 1982. It states a standardized p-value can be computed as **p = p[c] √(n/c),** where **n** is the sample size and **c** is a standardized sample size that **p[c]** is chosen against. The concern is that the real-time application is aiming at is for a “random number of sensor samples”. For simulation, considering the confidence interval as **95%**, and alpha level as **5%**, the cut-off would be approximately **1.645** based on the below formula in statistics. This implies that being 1.645 standard deviations away from zero implies entering the null hypothesis rejection region.

$$\text{cut\_off} = \text{norm.ppf}\ (1- 0.05) \tag{2}$$

$$\rho(\Psi) = \begin{cases} Ho\colon \Psi > \alpha\ -\ reject\ hypothesis, attack\ not\ detected \\ Ho\colon \Psi_{min} \le\ \ \Psi <\ \ \alpha - accept\ hypothesis, attack\ detected \end{cases} \tag{3}$$

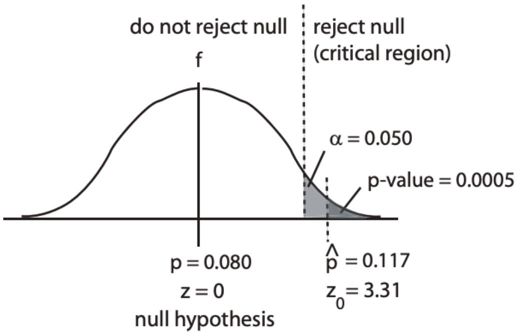


Figure 1: Single Tailed Function Analysis

Graphical justification of a one-tailed test in general and a right-tailed, on-tailed test in specific for the below mathematical equation is also provided in Figure 1.

3. RELATED GROUNDWORK – AN EXPERIMENTAL DISCUSSION

3.1. Exploring SL & USL Algorithms with NSL-KDD

It is important to understand how few traditional machine-learning-based supervised learning algorithms) behave in an anomalous environment before we proceed to discuss RL-based approaches. The authors also have worked with K-Means Clustering, Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA) and Autoencoders and recorded the metrics in TABLE II.

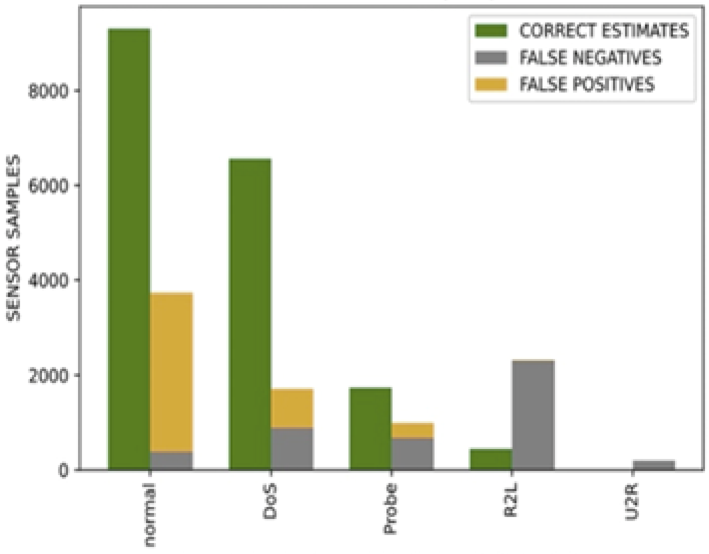
3.2. Graphical Overview of RL based Algorithms

The research work focuses on maximizing the trust/satisfaction metric based on Reinforcement Learning algorithms. RL helps the agent to learn from repeated trials and experiences in an interactive environment. Rewards and punishments mark the positive progressive behaviour and negative behaviour of the task respectively. All RL problems can be handled using Markovian Decision Processes (MDPs). Maximizing the reward, and minimizing the loss is the ultimate objective of a lucrative RL model. However, to be more specific, the fundamental goals of an agent are: (i) To maximize the average reward function, trust metric in this case (ii) To optimize latency (3) to reduce stopping time [28].

There exists a plethora of RL algorithms. For analysis of RL models, we consider the following three RL models – the basic Actor-Critic model, RL in a multi-agent adversarial environment and Modified Actor-Critic with one tailed function. The anomaly detection accuracy graphs are provided in Figure 2 for the NSL-KDD dataset.

TABLE II: Overview of SL and USL algorithms & their Results on NSL-KDD dataset

	Algorithm/ Classifier used	**Accuracy**	**Precision**	**Recall**	**F1 score**
Supervised Learning Techniques	*KNN*	0.99715	0.99678	0.99665	0.99672
	SVM	0.99371	0.99107	0.99450	0.99278
	Decision Trees	0.99662	0.99493	0.99732	0.99612
	Naïve Bayes	0.86733	0.98822	0.70308	0.82145
	Logistic Regression	0.99394	0.99093	0.99517	0.99305
	K-Means Clustering	0.99942	0.99884	0.99942	0.99913
Unsupervised Learning Techniques	*PCA*	0.68074	0.62274	0.68074	0.63210
	LDA	0.77629	0.78901	0.77629	0.77215
	Autoencoders	0.89069	0.88045	0.93493	0.90687
	QDA	0.55161	0.62075	0.55161	0.50604



Modified Actor Critic with one-tailed function

Figure 2: Results of few RL algorithms

4. METHODOLOGY

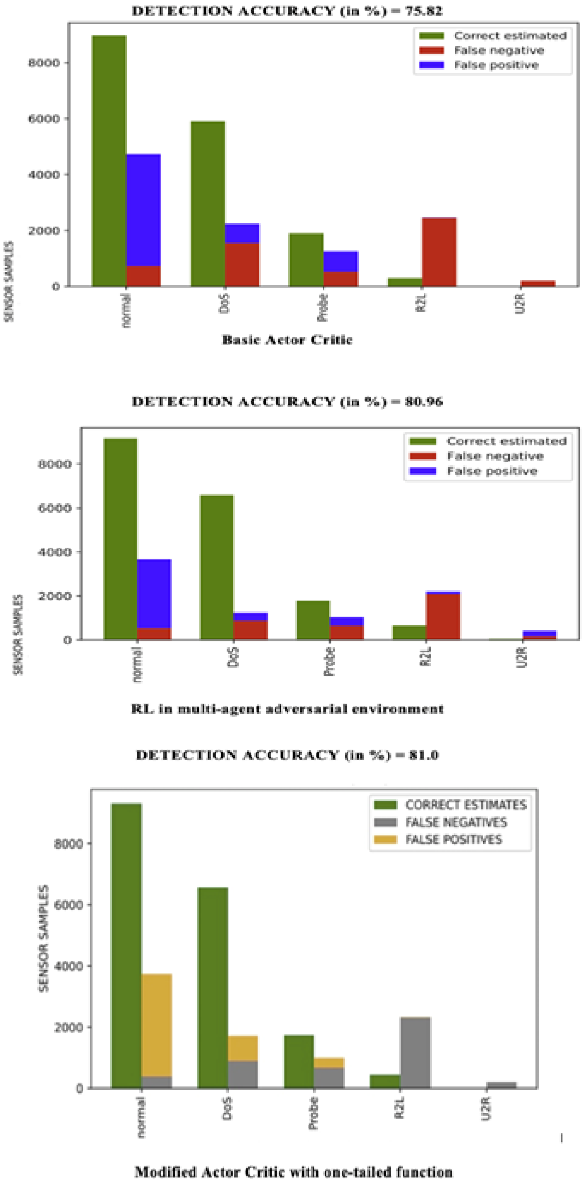
4.1. DL Algorithms Algorithm Suite – choice Strategy

Researchers have discussed the limitations of statistical and shallow machine learning methods and expressed that deep learning techniques are suitable to detect network attacks since these techniques are capable of executing both feature extraction and data classification. The map of DRL types is summarized in Figure 3. RL is considered to be one of the best solutions for IoT security since it banks on concurrent and corrective learning [29].

4.2. Literature Survey - Impact of RL Algorithms on Attack Detection

4.2.1. Basic RL algorithms

The authors in [30] propose an RL agent to observe the traffic. Another Q-DRL approach is proposed in [31] to monitor the sensory nodes. Partially Observable Markov decision process (POMDP) is projected in [32] to tackle anomaly detection problems. This model-free online workable RL approach fights attacks even without the previous knowledge of any other attack model. Actor Critic-based approach [33] helps learn a strategy which defends against attacks. The authors of [34] have highlighted RL-based work against DoS attacks.



Basic Actor Critic

RL in multi-agent adversarial environment

Modified Actor Critic with one-tailed function

Figure 2: Results of few RL algorithms

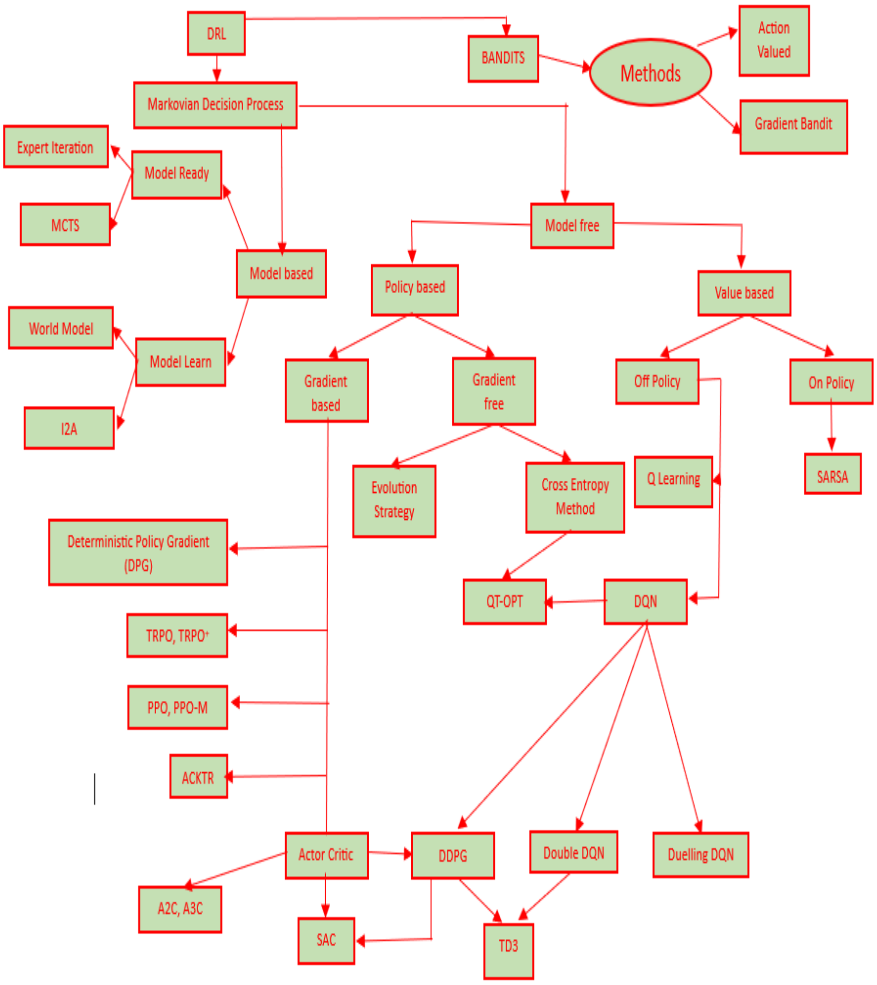


Figure 3: Classification of Deep Reinforcement Learning algorithms

The authors of [35] utilize a hypothesis test to determine whether a packet is sent from a particular source or not, and use the RL algorithm to find the value of the test threshold above which the packet gets certified as a “spoofed packet”. The updated state-action function computed by the receiver is used in reward calculation. Furthermore, in [36], the work based on Reachability & Inverse RL predicts and detect the assailed sensors. The authors used a CNN-based Deep Q-Network (DQN) implementation to design a power control scheme [37]. Mobility of secondary users across locations is a strategy which is used in [38] to manage jamming attack mitigation. For huge SINR values, there is a recursive CNN-based work [39] which the authors claim is capable of encountering the dynamically changing jamming patterns.

4.2.2. RL Actor-Critic Algorithms & its Variants

Basic AC methods are sensitive to perturbations in data. Asynchronous Advantage Actor Critic (A3C) has each of its workers loaded with a different set of weights contrasting to Advantage Actor Critic (A2C). Speed and robustness were promising. A3C provided parallel training of actor-critic but suffered optimal agent update problems which were later handled by A2C. Updates not happening immediately resulted in agents using older versions of parameters. [40] has its authors implement a model for anomaly detection based on A3C with an adaptable deep neural network for reward functions. The asynchronous workers model has put efforts to better the efficiency with the help of parallel computing. The authors of [41] proposed a DDQN & A3C coupled technique to convey the reduction of the number of simulation runs required to locate falsifying model inputs. The authors of [42] propose a classifier Adaptive Actor-Critic Neural network classifier to formulate an Intrusion Detection System (IDS). Another A3C-based IDS approach is highlighted in [43], automated network scan by service discovery.

The authors of [44] have detailed their work with Soft Actor Critic (SAC) based DRL for alert prioritization which aims to maximize rewards as well as entropy. SAC is a good performer however, it is complex in its implementation. A possible approach that we intend to follow in this paper is DDPG whose inputs are taken from the sensors through a LSTM memory layer. There is yet another SAC-based model [45] that enforces its attack detection policies with acceptable metric values of detection time, detection accuracy and energy consumed in the process. A compound action actor critic-based federated learning detection framework (CA2C – AFL) [46] discusses a selection strategy fused into the Asynchronous federated learning framework.

4.2.3. RL Policy Optimization Algorithms & its Variants

Trust Region Policy Optimization (TRPO) uses a surrogate function to learn complex policies. The Kullback-Leibler (KL) divergence objective of TRPO makes it difficult to implement as well. The authors of [47] have proposed a Proximal Policy Optimization (PPO) based intrusion detection hyperparameter control system (IDHCS) with a good F1 score of 0.96552 for the CICIDS2017 dataset. TRPO+ is a combination of TRPO and PPO code level optimizations. PPO-M refers to PPO without code level optimizations. Mikhail et.al. [48] discuss RL for attack mitigation in networks which revolves around DQN and PPO. The authors in [49] propose a PPO-based federated client selection scheme to optimize accuracy and system overhead as compared to their benchmark models.

4.2.4. RL Policy Gradient algorithms & its Variants

The training speed of PPO is impressive, however, Twin Delayed DDPG (TD3) has a much-elevated general performance and ability to transfer learning to other markets. As compared to DDPG, TD3 trains the agent with two Q-value functions. TD3 random noise component to next-state actions for smoothing while training a deterministic policy. TD3 completes the DDPG implementation with a smooth finish of clipped double learning, delayed policy updates and target policy smoothing. The authors of [50] have compiled the contributions of [51] which is DDPG based. Liu et al. have used the DDPG algorithm to train the agent to work against DDoS attacks and drop excess traffic overflood due to malicious data in SDNs. Wei et al. [52] in their work project the usage of DDPG to reclose transmission lines in cases of successful attacks. Sunghwan Kim et al. [53] propose a DDPG approach using real-time traffic analyzer monitoring results. The authors of [54] discuss a deep RL model to handle changing attack patterns which highlights good values of performance evaluation metrics. An upgrade of DDPG is accomplished as dynamic reward DDPG in [55] which shows 97.46% accuracy in detecting attackers. The authors of [56] propose a DDPG IDS approach to achieve a detection accuracy of 97.28% in the WUSTIL-IIOT-2021 test set.

4.3. Deep Deterministic Policy Gradient (DDPG) Framework

IDS can be classified as Learning-based mechanisms, Pattern-based mechanisms and Rule-based mechanisms [57]. The work discussed in the paper is based on IDS as a Learning-based mechanism. We use a DDPG approach which is model-free, policy-based and gradient-based for anomaly detection.

DDPG has the actor-network, critic network and replay memory. Both actor and critic have a dedicated target network for action evaluation and a current/ online network for action selection. Experience playback otherwise called memory replay is an added feature in DDPG.

The off-policy actor-critic algorithm learns a deterministic target policy from a exploratory behaviour policy to ensure adequate exploration. The neural networks compute action prediction for the current state and generate ID error at each step. The Current state acts as input to the action network, output will be an action from state space. Furthermore, the Q-value of the current state will be the critic's output. DDPG additionally supports an update rule to modify the weights of the actor-network. The obtained gradient will influence and update the critic network. The standard DDPG model with two separate neural networks for the actor and critic is shown in Figure 4. Deterministic modelling produces consistent outcomes for a given set of inputs, irrespective of the number of times the model is re-run or recalculated. One may notice the limitation of DDPG not fitting into a stochastic environment, unlike the SAC model. However, feeding inputs to the DDPG model through an LSTM network would make things better for data exploration. Overfitting limitations also can be handled with the help of auto encoders, ensemble, regularization, feature selection, cross-validation, increasing percentage of training data and additive noise in data.

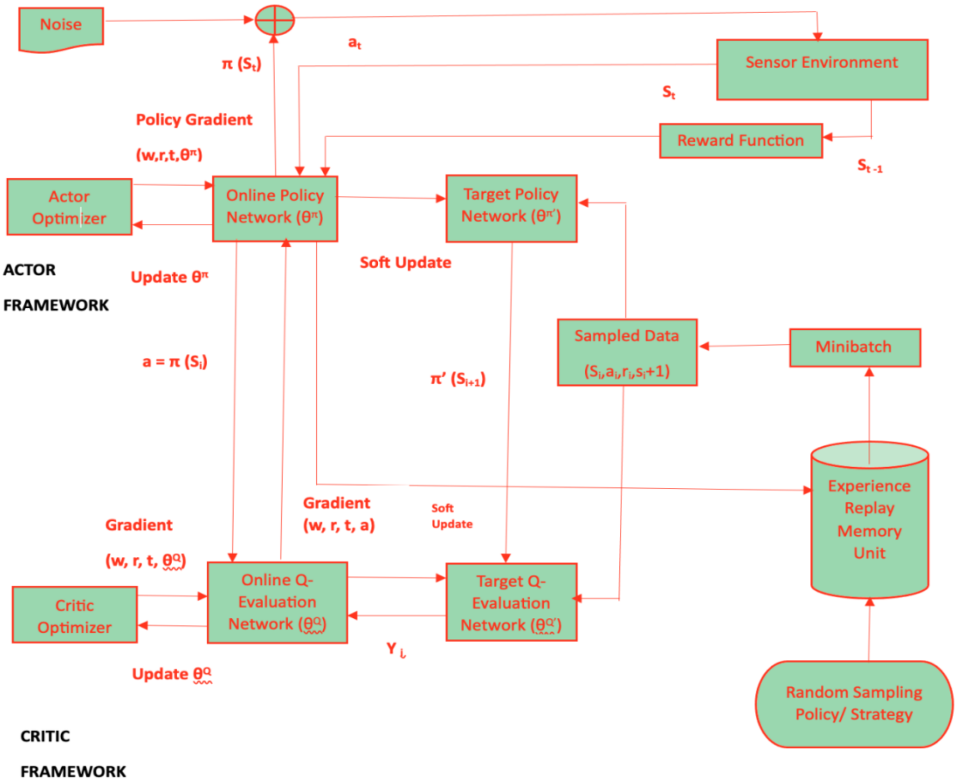


Figure 4: DDPG Framework

5. System Model & Design

5.1. Problem Formulation

Let us consider the current state to be say **s_t**, which belongs to the state space S(t).All possible states are associated with a hypothesis highlighted in section 2.4. Posterior probabilities and posterior trust value Ψ(t) are computed based on the hypothesis H_j at time (t). The observation information with the agent at any time (t) is given by:

$$O_t \subseteq [Pr]; \qquad t: 1 \text{ to } \tau\text{-}1 \tag{4}$$

Agent adopts a sequence of actions depending on the critic's feedback, mathematically represented as:

$$At \subseteq [P_r]; \qquad t: 1 \text{ to } \tau\text{-}1 \tag{5}$$

The trust vector can be expressed either as the probability of the state being '0', the posterior probability that the ith process is non-anomalous or as the probability of hypothesis $H_j$being true at time (t).

$$\Psi i(t) = \rho\ (si = 0 \mid Ot, At); \qquad t: 1 \text{ to } \tau\text{-}1 \tag{6}$$

$$\Psi i(t) = \rho\ (H = j \mid Ot, At); \qquad t: 1 \text{ to } \tau\text{-}1 \tag{7}$$

Bayes rule is used to handle samples in real time. The probabilities are updated based on the sequence of actions. Bayes rule is formulated as:

$$\Psi i(t) = \frac{\rho\ (\mathrm{H} = \mathrm{j}).\rho\ (\mathrm{Z}[\mathrm{A}(\mathrm{t})] \mid (\mathrm{H}{=}\mathrm{j})}{\sum_{j=1}^{H} (H{=}j).\rho(\mathrm{Z}[\mathrm{A}(\mathrm{t})] | (\mathrm{H}{=}\mathrm{j})} \tag{8}$$

Now that we have considered the confidence interval, the design must ensure to abide by the defined confidence interval margins and not hop over the interval. Logit transformation can be used to quantify confidence levels [58]. Trust metrics and confidence intervals influence reward maximization. The trust metric is the Bayesian log-likelihood ratio of the hypothesis at time (t) given as:

$$\xi_j\ (\Psi) = \log \frac{\Psi(\mathrm{j})}{1-\Psi(\mathrm{j})} \tag{9}$$

The average Bayesian log-likelihood ratio is represented as below:

$$\xi_{avg}\ (\Psi) = \sum_{j=1}^{H} \xi j(\Psi).\ \Psi_j \tag{10}$$

The instantaneous reward of the MDP is given by:

$$r_{\pi(t)} = \xi avg\ (\Psi(t)) - \xi avg\ (\Psi(t - 1)) \tag{11}$$

We can further use $r_{\pi(t)}$ to average the reward components.

$$R\pi\ (t) \ = \frac{1}{\tau} \sum_{t\ =\ 1}^{\tau\ -1} E^{\pi}[\ r_{\pi}\ (t)] \tag{12}$$

The asymptotic expected reward is based on the average rate of increase in the confidence level on the true hypothesis H and is defined as :

$$Rt\ (st, at) = R(\pi) := \lim_{O_\tau \to \infty} \frac{1}{O_\tau} E^{\Psi}\ [\Im(\Psi(O_\tau\ +\ 1)\ -\ \Im(\Psi(1)] \tag{13}$$

The DDPG algorithm has a framework wherein the agent/ actor takes the current state as the information from the observation space, the environment. Accordingly, the actor performs a particular action based on the defined policy π. θ_π will be the network parameter of the policy π.

$a_t \in A = \pi\ (\Psi(t\text{-}1)) + \eta t$ (stochastic noise component) (14)

Also, the deterministic policy gradient models the policy as a deterministic decision. Therefore, we can also write;

$a_t \in A = \pi(\ st, \theta_\pi\ ) + \eta t$ (15)

The Critic network is optimized, and its parameters are updated by the difference between the two networks.The loss function is shown below.

$L(\theta_Q) = \frac{1}{N} \sum_{i=t}^{N} y_i - Q\ (S_i, a_i, \theta_Q\ ))\ ^2$ (16)

The stochastic policy gradient concept of DDPG aims at adjusting the network parameter weights θ of the policy π in the direction of the performance gradient $\nabla\theta_\pi\ J(\pi)$.

The policy gradient does not depend upon the gradient of the state distribution even when it is factual that the state distribution $\rho\pi$ (s) depends on the policy parameters [59].

$\nabla\theta_\pi\ J\ (\pi) = E\ s \sim \rho_\pi,\ a \sim \pi_\theta\ [\nabla_\theta \log \pi_\theta\ (a|s).Q_\pi\ (s,a)]$ (17)

$\nabla\theta_\pi\ J\ (\pi) = (1/N)\ \{\sum_{i=t}^{N} \nabla a\ Q(s,a\ |\ \theta_Q\ )\ |\ s{=}s_i,\ a = \pi(s_i)\ \}\ X\ \ \{\ \nabla_{\theta\pi}\ .\ \pi\ (s,\theta_\pi\ )\ |\ s{=}s_i\ \}$ (18)

The target policy network and target Q-network will be updated by using the respective online policy network and online Q-networks. The update equations are mentioned below:

$\theta_Q' \leftarrow \alpha\theta_Q + (1\text{-}\alpha)\ \theta_Q'$
$\theta_\pi' \leftarrow \alpha\theta_\pi + (1\text{-}\alpha)\ \theta_\pi'$ (19)

It is to be noted that α is called the update coefficient which is usually small-valued to slow down the target. Hence, it is also termed as SOFT update coefficient. Typical values can be 0.1 or 0.01.

5.2. LSTM-based Network Architecture for DDPG Implementation

DDPG is deterministic and a complex algorithm like SAC which is inherently stochastic is not being used in our work. A possible approach is to use a neural network with a sequential information structure which can learn from long-term dependencies. The wrap-up,recurrent connections in RNNs aid the network in storing past information and hence handling temporal dependency issues. The loops in the layer connections store the state value and envision the sequential inputs. However, the vanishing gradient problem in RNNs during back propagation eye for a superior network called LSTM which is eventually a stack of memory cells.

LSTM networks have memory blocks connected into layers instead of neurons. The memory cell constitutes 3 important gates – input, output and forget gates.

(a) The "**forget**" gate determines what details are to be discarded from the cell state block with the help of the **sigmoid function.** it looks at the previous state(**h_{t-2}**) and the content input (**X_{t-}**

$_{1}$) and outputs a number between 0 (to eliminate) and 1 (to retain) for each value in the cell state $\mathbf{C}_{t-2}$.

$$\begin{aligned} f_{t-1} &= \sigma (W_f . [h_{t-2}, x_{t-1}] + b_f) \\ &= \sigma (W_{sf} . x_{t-1} + W_{hf} . h_{t-2}) + b_f) \end{aligned} \tag{20}$$

(b) The “input” gate layer determines which value from input should be used to further do modifications in the cell state. This is followed by a “tanh” layer to create a vector of new candidate or potential nominee values $C^{\sim}_{t-1}$to be included in the state. The cell state will be later updated to C_{t-1} with the help of C_{t-2}, $f_{t-1}$and i_{t-1}.

$$\begin{aligned} i_{t-1} &= \sigma (W_i . [h_{t-2}, x_{t-1}] + b_i) \\ &= \sigma (W_{ih} . h_{t-2} + W_{ix} . x_{t-1}) + b_i) \end{aligned} \tag{21}$$

$$C^{\sim}_{t-1} = \tanh (W_c [h_{t-2}, x_{t-1}] + b_c)$$

$$= \tanh (W_{ch} . h_{t-2} + W_{cx} . x_{t-1} + b_c) \tag{22}$$

$$C_{t-1} = f_{t-1} * C_{t-2} + i_{t-1} * C^{\sim}_{t-1} \tag{23}$$

(c) The final gate is the “output” gate layer. A sigmoid layer checks, decides and what sections of the cell state will be redirected to the output. The system be implementing a cell state to the tanh function, and multiply it with the sigmoid gate output.

$$\begin{aligned} o_{t-1} &= \sigma (W_o [h_{t-2}, x_{t-1}] + b_o) \\ &= \sigma (W_{oh} . h_{t-2} + W_{ox} . x_{t-1} + b_o) \end{aligned} \tag{24}$$

$$h_{t-1} = o_{t-1} * \tanh (C_{t-1}) \tag{25}$$

To make understanding and reference equations easier, a tabulation of all used symbols corresponding to LSTM is provided in TABLE III. The internal structure an LSTM cell depicting all three gates is shown in Figure 5. Also, LSTM implementations are based on minimalistic pre-processing. These models can also perform on sequential time series data to identify anomalies sometimes even without dimensionality reduction techniques. The collected sensor samples $z_0, \ldots , z_{t-1}$) are input into the LSTM neural network to extract the features, z_t’ including the desired features favourable to detecting anomalies. The detection model along with the LSTM-based neural network in Figure 6 depicts a layered view of an input layer, four LSTM cascades, a dense layer of 512 neurons and a Softmax output layer.

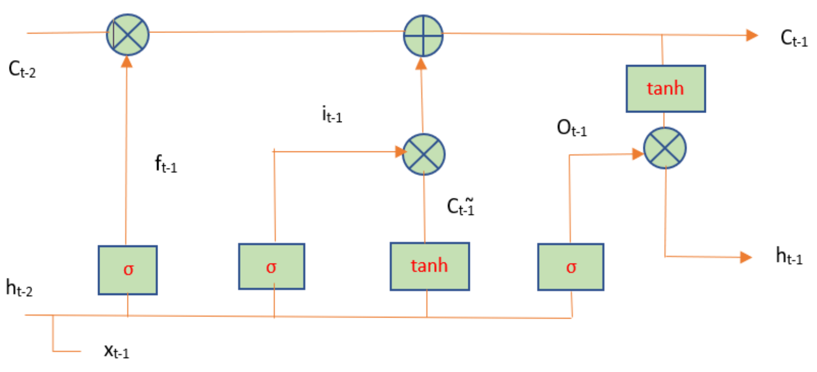


Figure 5: The LSTM Cell

TABLE III: Abbreviations used in LSTM

SYMBOL	UNDERSTANDING	REFERRED GATE
f_{t-1}	Forgot gate	Forgot
σ	Sigmoid function	All gates
i_{t-1}	Input gate	Input
o_{t-1}	Output gate	Output
$W(f,i,C,o)$	Weight matrix for respective gates	All gates
h_{t-2}	Output of previous LSTM block	All gates
C_{t-2}	LSTM previous memory content	Candidate values in input gate
$\tilde{C}_{t-1}$	LSTM current memory contents	Candidate values in input gate
C_{t-1}	LSTM new memory contents	Candidate values in input gate
$b(f,C,o)$	Biases for respective gates	All gates
x_{t-1}	Current input	All gates

We have taken into account the number of features (numb=42) available at the input and created a (numb x_1) input vector. The single input layer will receive the data (legitimate + attacks) with 42 features. A 42 x 1 input matrix or input vector will be formulated to fit the best of the 42 features. Non-numeric features are avoided by label encoding them into numeric features. Input data has also been one-hot encoded as binary vectors. The input dataset matrix, in its pre-processing stages, gets split into training and testing datasets, and one-hot encoding techniques have been used. The pre-processed data as input for LSTM. 512 units are used at each LSTM layer. The proposed model uses 2 LSTM layers and a timestep maintained at 4, a typical 4 times unroll. Therefore, the set of equations ranging from 20 to 25 will be computed four times for each timestep. However, the weight matrices and biases are used once in common for all timesteps since they are not time-dependent. A leaky ReLU activation layer is used to support accelerated learning. Normalization decreases error rates. Regularization (L2) layers help mitigate the effect of overfitting in our model. The output layer determines whether an anomaly has been detected or not. There will be no changes in neuron weights during backpropagation, system is stable.

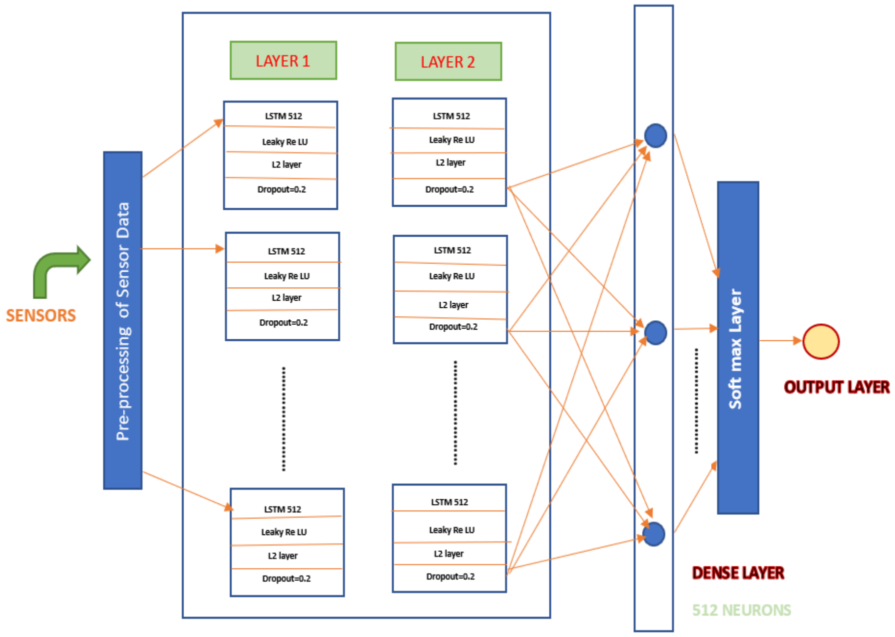


Figure 6: LSTM Model Overview

6. Algorithm Overview

Preliminary Initialization:

```
Initialization of Online Actor/ Policy Network: $\pi ( S_t | \theta^{\pi})$
Initialization of Online Critic/ Q Network: $Q ( S_t, a | C^{Q})$
Weights: $\theta^{\pi} , \theta^{Q}$
Initialize target policy network and target Q network using online network parameters:
$\theta^{Q'} \leftarrow \theta^{Q}$
$\theta^{\pi'} \leftarrow \theta^{\pi}$
Initialize the Replay Buffer
Core Steps:
forepisode_index Ep = 1, 2, 3,….do
    Set time_index t = 1
    Sample data $(Z_0,Z_1,….Z_{t-1})$, enter LSTM network to give $Z_t$'.
    Generate hypothesis (H) to be true according to a range of $\Psi$.
    while $\Psi_{min} \leq \Psi < \alpha$do

        • Actor-network selects action according to decision policy:
        ▪ $a_t \in A = \pi (St' | \theta^{\pi} ) + \eta_t$(stochastic noise)
        • Observe reward $r_t$ and next state $S_{t+1}$'
        • Store ($S_t$', $a_t$, $r_t$, $S_{t+1}$') in Replay Buffer
        • if Buffer size > Minibatch size then

            ○ Sample (Z) from Buffer.
            ○ Reward calculation based on confidence interval:
            ○ $R(\pi) = \lim_{O_\tau \to \infty} \frac{1}{O_\tau} E^{\Psi} [\mathfrak{I}(\Psi(O_\tau + 1) - \mathfrak{I}(\Psi(1)]$
            ○ Update critic network with minimized TD error:

Loss $L (\theta^{Q}) = \sum_{i=1}^{\#Z} [ Ri (\pi) - Q (S_i, a_i, \theta^{Q}) ]^2$

        • Update actor-network:

                $\nabla_{\theta}^{\pi} J (\pi) = \sum_{t=1}^{\#Z} \nabla_a Q (s,a|\theta^{Q}) |_{S=Si, a = \pi(Si)} \} . \nabla_{\theta}^{\pi} . \pi (s,\theta^{\pi} ) |_{S=Si}$
                                        end if

end while

    update target networks by using the updated networks, take $\alpha$ = 0.005.

                    $\theta^{Q'} \leftarrow \alpha\theta^{Q} + (1-\alpha) \theta^{Q'}$
                    $\theta^{\pi'} \leftarrow \alpha\theta^{\pi} + (1-\alpha) \theta^{\pi'}$
            Finalize hypothesis status (anomaly detection status)
                Accept hypothesis (1) – Attack detected
                Reject hypothesis (0) – No attack detected
end for
```

7. Results and Discussion

7.1. Underlying Neural Network

We make a comparative analysis using RNN as well as LSTM along with their variants. The best model is chosen for model implementation to co-work with the DDPG-based algorithm. A comparative analysis of average values of accuracy is done alongside the number of epochs. For experimental study, we have chosen a train of 15 epochs, an optimal batch size of 32 and a validation split of the data as 0.33. Figure 7 provides a bar depiction of the result. The results have motivated the authors to proceed with LSTM1 as the base network model. The metric of accuracy has been used to determine the choice of LSTM in general or recurring LSTMs in specific over RNN.

7.2. Selection of Dataset and Hyperparameters

This research work makes use of NSL-KDD to compare our model with different intrusion detection models and frameworks. The workable ratio of training and testing data is taken to be approximately 67% and 33% respectively. Both training and testing datasets have 42 features which are also the inputs to the model. The dataset is being divided into separate datasets for each of the categories namely Normal, Denial-of-service (DoS), Probe, Remote-to-Local (R2L) and User-to-Root (U2R) attacks.

RNN 1 model is a simple RNN with a learning rate of 0.01, an Adam optimizer, a sigmoid activation function and 80 hidden nodes. The next model namely RNN 2 has a modification concerning hidden nodes being a 100. The rest of the parameters remain the same. LSTM 1 model uses an LSTM cascade with 512 neurons aided with the Leaky Relu activation function. The model uses 2 LSTM layers with dropout maintained at 20%, a single dense activation layer and one Softmax output layer. Also, each LSTM layer contains 80 hidden nodes.The final model analysed LSTM 2 has a variation of a number of hidden layers and activation function as compared to LSTM 1.

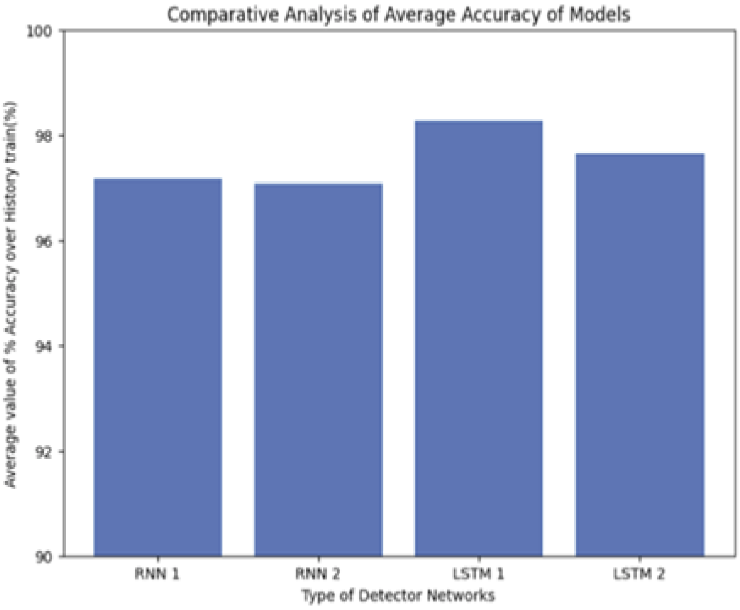


Figure 7: Comparative Analysis of RNN and LSTM Models

TABLE IV: Hyperparameters with values

Hyperparameters	Values	Hyperparameters	Values
Mini Batch size	32	Learning Rate of Critic network	0.002
Activation functions	Leaky ReLu, Softmax	Episode Count	50
Optimizer	Adam	Neuron dropout	0.2
Loss Function	MSE	Replay Buffer size	50000
Discount factor	0.99	Soft target update tau	0.005
Learning Rate of Actor network	0.0001		

7.3. Metrics of Evaluation & Rewards Tally

The variation of the values of Rewards concerning the number of epochs or episodes is shown in Figure 8. DDPG is purely Reinforcement Learning and finding the reward function is challenging, and depends on continuous state space. Figure 8 shows a reward tally of conventional DDPG versus DDPG-BN model. The proposed model reward calculation is different from its counterpart. The calculation is purely based on confidence interval.

TABLE V: Performance Evaluation of Anomaly Detection models using NSL-KDD

Reference Title	Fundamental concept used	Accuracy	F1 score
Actor Critic Approach based Anomaly Detection for Edge Computing Environments [25]	Actor Critic	81	-
A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks [60]	RNN-IDS	83.28	-
Application of Improved Asynchronous Advantage Actor Critic Reinforcement Learning Model on Anomaly Detection [40]	A2C	79.7	84.63
PSO-Driven Feature Selection and Hybrid Ensemble for Network Anomaly Detection [61]	feature selection with a hybrid ensemble approach	90.39	90.7
Network intrusion detection based on novel feature selection model and various recurrent neural networks [62]	hybrid Sequence Forward Selection (SFS) algorithm and Decision Tree (DT) model	96.9	-
Wireless senor network intrusion detection system based on MK-ELM [63]	Multi Kernel Extreme Learning Machine (MK-ELM)	98.34	-
Building an Effective Intrusion Detection System Using the Modified Density Peak Clustering Algorithm and Deep Belief Networks [64]	modified density peak clustering algorithm (MDPCA) and deep belief networks (DBNs)- MDPCA-DBN	82.08	81.75

Attention based multi-agent intrusion detection systems using reinforcement Learning [65]	Deep Q-Network logic in multiple distributed agents & attention mechanisms	97.2	97.8
Application of deep reinforcement learning to intrusion detection for supervised problems [66]	DDQN	89.78	91.02
	DQN	87.87	
	Policy gradient	78.73	79.09
	Actor Critic	80.78	81.11
GAN-based imbalanced data intrusion detection system [67]	Adversarial environment Reinforcement Learning (AE-RL)	80.16	79.4
A context-aware robust intrusion detection system: a reinforcement learning-based approach [68]	DQN context aware	81.8	-
Proposed DDPG-BN	DDPG based	98.37	85.22

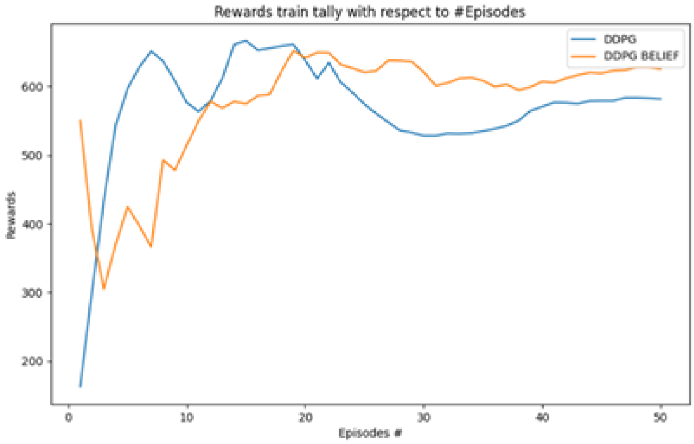


Figure 8: Rewards Tally of Proposed Model

A set of vital model evaluation metrics has been graphically analysed to document the performance of the DDPG-BN model. Refer to Figure 9. The results witness a noticeable improvement in rewards with the increase in episode number.

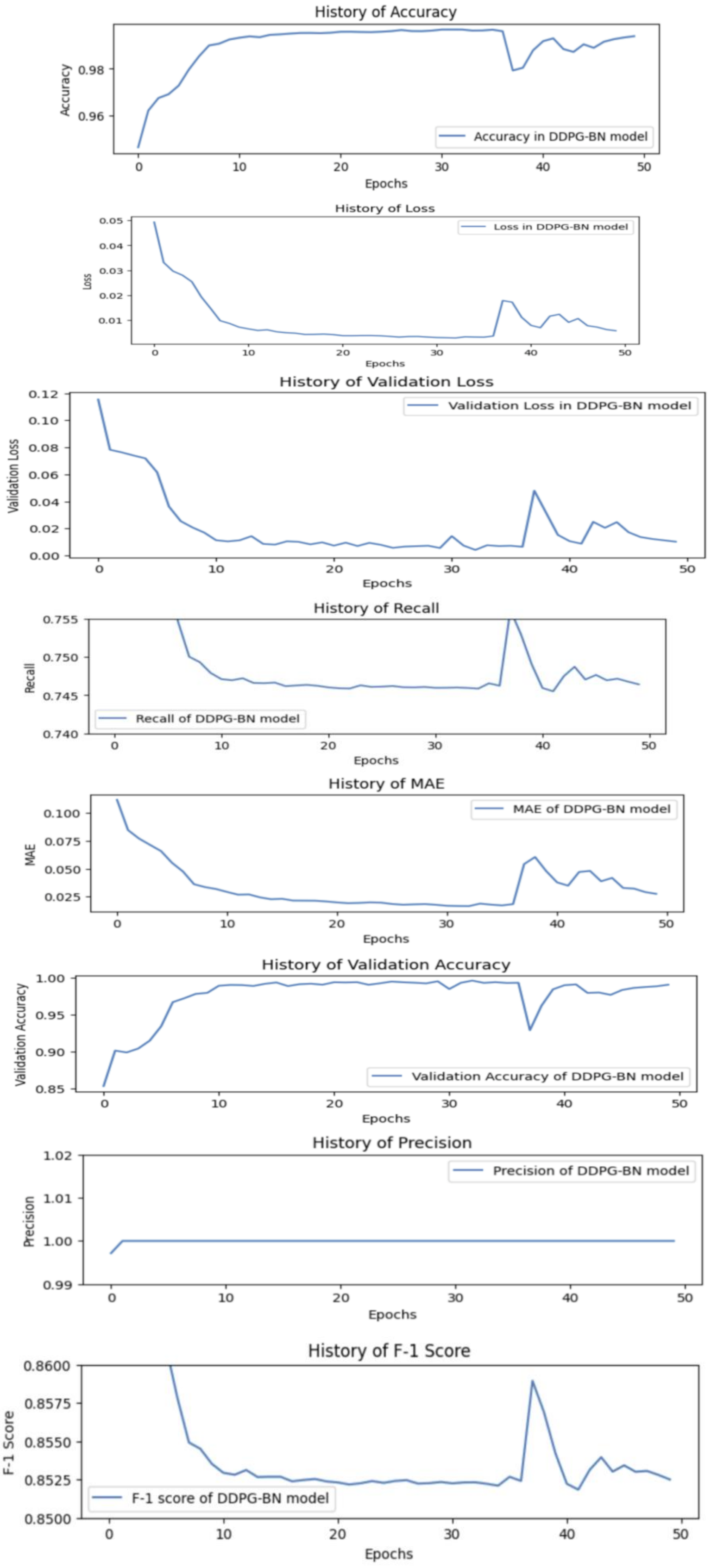


Figure 9: Comparative Analysis of Evaluation Metrics – DDPG-BN model

8. CONCLUSION AND FUTURE WORK

The proposed work implements the LSTM-based DDPG algorithm for anomaly detection. LSTM framework has proven to be effective for temporal characteristic data. The work aims at providing an attack detection model or otherwise an Intrusion Detection model with relatively good evaluation metrics as compared to its counterpart models. The reward calculations are purely based on confidence intervals. We have banked upon the Trust metric and confidence interval to be impacting reward maximization. The trust metric we have considered is the Bayesian log-likelihood ratio of the hypothesis. The work demonstrates the DDPG-BN algorithm to iterate the temporal dataset chosen to demonstrate the validity of the reward function. The proposed model showcases a generic authentication protocol and finds its applications in detecting attacks in edge devices like sensor devices, actuators or even router switches and gateways. Domain-specific use includes the oil & gas industry, in-hospital data monitoring, autonomous driving, generic traffic management and even simple smart homing mechanisms.

The results demonstrate that the reward values fluctuate between bad and good values as compared to the basic DDPG algorithm up to a few initial episodes of the exploratory stage. Later on, the learning curve becomes steeper. The proposed DRL approach in our work DDPG-BN provides an average detection accuracy of around 98.37 %. The proposed model performed better than the conventional Actor-Critic model and few other conventional ML model contributions by other researchers. However, the work is confined to the binary classification of attacks on a single dataset. Future work can be aligned to multiple datasets for detecting anomalies. Also, the use of ensemble classifiers and autoencoders in the design may bring in better reward values and valuable metric information. A stacking model [69] with classifiers, encoders and ensemble techniques can favour as an add-on to the model.

9. CONFLICTS OF INTEREST

The authors declare no conflict of interest. If you have any conflict of interest, let me know.

AUTHORS

Shruthi. N is a Ph.D. research scholar in Bangalore, Karnataka, India. She received a Bachelor's degree in Electronics Communication Engineering and a Master's degree in Digital Electronics Communication in 2005 and 2014 respectively. Her areas of interest are Network Security, IoT and Embedded Systems. She has nearly 4 years of industry experience and 8.5 years of teaching experience with 7 International Journal publications to her credit.

Dr.Siddesh.G.K. is the Head of the ECE Department, at ALVA's Institute of Engineering & Technology. He received a Bachelor's degree in Electronics Communication Engineering from Bangalore University in 1998, an M.Tech. in Digital Electronics and Advanced Communications from Manipal Institute of Technology, Manipal, Karnataka in 2002 and a Ph.D.in Electronics Communication Engineering from Visvesvaraya Technological University, Belagavi in 2013. His work experience includes academic, and research administration of more than 20+ years in various engineering colleges. He has published more than 45 research papers in various National and international Journals and Conferences in India and abroad. He also has book chapters from reputed publishers to his credit.

REFERENCES

[1] J. Zhang, B. Chen, Y. Zhao, X. Cheng, and F. Hu, "Data security and privacy-preserving in edge computing paradigm: Survey and open issues," IEEE Access, vol. 6, pp. 18209–18237, 2018.

[2] H. Yang, A. Alphonse, Z. Xiong, D. Niyato, J. Zhao, and K. Wu, "Artificial-intelligence-enabled intelligent 6G networks," IEEE Netw., vol. 34, no. 6, pp. 272–280, Nov./Dec. 2020.

[3] Zhang, Y.; Cheng, Y. An Amplification DDoS Attack Defence Mechanism using Reinforcement Learning. In Proceedings of the 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Leicester, UK, 19–23 August 2019; pp. 634–639.

[4] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine learning in iot security: Current solutions and future challenges," IEEE Communications Surveys & Tutorials, vol. 22, no. 3, pp. 1686–1721, 2020.

[5] Yinhao Xiao, Yizhen Jia, Chunchi Liu, Xiuzhen Cheng, Fellow, IEEE, Jiguo Yu, Senior Member, IEEE, and WeifengLv, "Edge Computing Security: State-of-The-Art and Challenges", IEEE Xplore, 2019. DOI: 10.1109/JPROC.2019.2918437.

[6] "Financial impact of Mirai DDoS attack on dyn revealed in new data," https://www.corero.com/blog/797-financial-impact-of-miraiddos-attack-on-dyn-revealed-in-new-data.html, 2017.

[7] H. Luo, Y. Lin, H. Zhang, and M. Zukerman, "Preventing DDoS attacks by identifier/locator separation," IEEE Network, vol. 27, no. 6, pp. 60– 65, 2013.

[8] R. Xu, W. Ma, and W. Zheng, "Defending against udp flooding by negative selection algorithm based on eigenvalue sets," in 2009 Fifth International Conference on Information Assurance and Security, vol. 2, Aug 2009, pp. 342–345.

[9] J. Mirkovic, G. Prier, and P. Reiher, "Attacking ddos at the source," in Network Protocols, 2002. Proceedings. 10th IEEE International Conference on. IEEE, 2002, pp. 312–321.

[10] X. Xu, Y. Sun, and Z. Huang, "Defending DDoS attacks using hidden markov models and cooperative reinforcement learning," in Pacific-Asia Workshop on Intelligence and Security Informatics. Springer, 2007, pp. 196–207.

[11] T. Shon, Y. Kim, C. Lee, and J. Moon, "A machine learning framework for network anomaly detection using SVM and ga," in Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop. IEEE, 2005, pp. 176–183.

[12] T. Frassetto, P. Jauernig, C. Liebchen, and A.-R. Sadeghi, "IMIX: In-process memory isolation extension," in 27th USENIX Security Symposium (USENIX Security 18). Baltimore, MD: USENIX Association, 2018, pp. 83–97.

[13] S. Shirali-Shahreza and Y. Ganjali, "Protecting home user devices with an sdn-based firewall," IEEE Transactions on Consumer Electronics, vol. 64, no. 1, pp. 92–100, Feb 2018.

[14] C. Dietz, R. L. Castro, J. Steinberger, C. Wilczak, M. Antzek, A. Sperotto, and A. Pras, "Iot-botnet detection and isolation by access routers," in 2018 9th International Conference on the Network of the Future (NOF), Nov 2018, pp. 88–95.

[15] P. A. R. Kumar and S. Selvakumar, "Distributed denial of service attack detection using an ensemble of neural classifier," Computer Communications, vol. 34, no. 11, pp. 1328–1341, 2011.

[16] Xiaoyong Yuan, Chuanhuang Li, Xiaolin Li, "DeepDefense: Identifying DDoS Attack via Deep Learning",IEEE, 2017.

[17] Dinh Thi Thai Mai et al., "DDOS ATTACKS DETECTION USING DYNAMIC ENTROPY INSOFTWARE-DEFINED NETWORK PRACTICAL ENVIRONMENT", International Journal of Computer Networks & Communications (IJCNC) Vol.15, No.3, May 2023 DOI: 10.5121/ijcnc.2023.15307.

[18] K. Ross, M. Moh, T.-S. Moh, and J. Yao, "Multi-source data analysis and evaluation of machine learning techniques for SQL injection detection," in Proceedings of the ACMSE 2018 Conference, ser. ACMSE '18. New York, NY, USA: ACM, 2018, pp. 1:1–1:8. [Online]. Available: http://doi.acm.org/10.1145/3190645.3190670.

[19] S. Rathore, P. K. Sharma, and J. H. Park, "Xssclassifier: An efficient XSS attack detection approach based on machine learning classifier on sss." Journal of Information Processing Systems, vol. 13, no. 4, 2017.

[20] Roy, S. Setty, A. Kilzer, V. Shmatikov, and E. Witchel, "Airavat: Security and privacy for Map Reduce," in Symposium on Networked Systems Design and Implementation (NSDI). USENIX - Advanced Computing Systems Association, April 2010. [Online]. Available: https://www.microsoft.com/enus/research/publication/airavat-security-and-privacy-for-mapreduce/

[21] C. Liu et al., "A New Deep Learning-Based Food Recognition System for Dietary Assessment on an Edge Computing Service Infrastructure," IEEE Trans. Services Computing. DOI: 10.1109/TSC.2017.2662008.

[22] Prashanth Subramaniam, Maninder Jeet Kaur," Review of Security in Mobile Edge Computing with Deep Learning", Advances in Science and Engineering Technology International Conferences (ASET), 2019, DOI: 10.1109/ICASET.2019.8714349.

[23] Li, H., Ota, K., & Dong, M. (2018). Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing. IEEE Network, 32(1), 96–101. doi:10.1109/mnet.2018.1700202.

[24] Yuanfang Chen, Yan Zhang, Sabita Maharjan, "Deep Learning for Secure Mobile Edge Computing", 23 Sep 2017, arXiv:1709.08025v1 [cs.CR].

[25] Shruthi N, Siddesh G K," Actor-Critic Approach based Anomaly Detection for Edge Computing Environments", International Journal of Computer Networks & Communications (IJCNC) Vol.15, No.1, January 2023.

[26] Sander Greenland et al., "Statistical tests, P values, confidence intervals, and power: a guide to misinterpretations", European Journal of Epidemiology, May 21, 2016, 31: 337–350.

[27] H. Chernoff, "Sequential design of experiments," The Annals of Mathematical Statistics, vol. 30, no. 3, pp. 755–770, 1959.

[28] C. Zhong, M. C. Gursoy, and S. Velipasalar, "Deep actor-critic reinforcement learning for anomaly detection," in 2019 IEEE Global Communications Conference (GLOBECOM), pp. 1–6, IEEE, 2019.

[29] G. Caminero, M. Lopez-Martin, and B. Carro, "Adversarial environment reinforcement learning algorithm for intrusion detection," Computer Networks, vol. 159, pp. 96–109, 2019.

[30] Erhan, D.; Anarım, E. Bo˘ gaziçi University distributed denial of service dataset. Data Brief 2020, 32, 106187. [CrossRef] [PubMed]

[31] Jokar, P.; Leung, V.C.M. Intrusion Detection and Prevention for ZigBee-Based Home Area Networks in Smart Grids. IEEE Trans. Smart Grid 2018, 9, 1800–1811.

[32] Kurt, M.N.; Ogundijo, O.; Li, C.; Wang, X. Online Cyber-Attack Detection in Smart Grid: A Reinforcement Learning Approach. IEEE Trans. Smart Grid 2019, 10, 5174–5185.

[33] Feng, M.; Xu, H. Deep reinforcement learning based optimal defence for cyber-physical system in the presence of unknown the cyber-attack. In Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, USA, 27 November–1 December 2017; pp. 1–8.

[34] Aashma Uprety and Danda B. Rawat," Reinforcement Learning for IoT Security: A Comprehensive Survey", Y IEEE INTERNET OF THINGS JOURNAL, EARLY ACCESS DOI LINK: HTTPS://DOI.ORG/10.1109/JIOT.2020.3040957, Feb 2021.

[35] J. Liu, L. Xiao, G. Liu, and Y. Zhao, "Active authentication with reinforcement learning based on ambient radio signals," Multimedia Tools and Applications, vol. 76, no. 3, pp. 3979–3998, 2017.

[36] N. Bezzo, "Predicting malicious intention in cps under cyber-attack," in 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS), pp. 351–352, IEEE, 2018.

[37] Y. Chen, Y. Li, D. Xu, and L. Xiao, "Dqn-based power control for IoT transmission against jamming," in 2018 IEEE 87th Vehicular Technology Conference (VTC Spring), pp. 1–5, 2018.

[38] L. Xiao, X. Wan, W. Su, Y. Tang, et al., "Anti-jamming underwater transmission with mobility and learning,

[39] X. Liu, Y. Xu, L. Jia, Q. Wu, and A. Anpalagan, "Anti-jamming communications using spectrum waterfall: A deep reinforcement learning approach," IEEE Communications Letters, vol. 22, no. 5, pp. 998–1001, 2018.

[40] Kun Zhou et al.," Application of Improved Asynchronous Advantage Actor-Critic Reinforcement Learning Model on Anomaly Detection", 25 February 2021, Entropy 2021, 23, 274. https://doi.org/10.3390/ e23030274.

[41] Takumi Akazaki et al., "Falsification of Cyber-Physical Systems Using Deep Reinforcement Learning", Springer, International Symposium on Formal Methods, pp 456-465, July 2018.

[42] R.Sudhakar et al.," Novel Probabilistic Clustering with Adaptive Actor-Critic Neural Network (AACN) for Intrusion Detection Techniques", Advances in Intelligent Systems and Computing, Emerging Research in Data Engineering Systems and Computer Communications Proceedings of CCODE 2019, pp 561-566.

[43] Eric Muhati et al.," Asynchronous Advantage Actor-Critic (A3C) Learning for Cognitive Network Security", 14 April 2022, IEEE, 10.1109/TPSISA52974.2021.00012.

[44] Lalitha Chavali, Tanay Gupta, Paresh Saxena, "SAC-AP: Soft Actor Critic based Deep Reinforcement Learning for Alert Prioritization", 2022 IEEE Congress on Evolutionary Computation (CEC), IEEE, 06 September 2022, DOI: 10.1109/CEC55065.2022.9870423.

[45] Bhargavi K et al.," Man-in-the-Middle attack Explainer for Fog Computing using Soft Actor Critic Q-Learning Approach", 2022 IEEE World AI IoT Congress (AIIoT), 13 July 2022, IEEE, 10.1109/AIIoT54504.2022.9817151.

[46] Weili Wang et al.," A VHetNet-Enabled Asynchronous Federated Learning-Based Anomaly Detection Framework for Ubiquitous IoT", 6 March 2023,

[47] arXiv:2303.02948 [cs.NI].

[48] Hyun Han, Hyukho Kim, Yangwoo Kim, "An Efficient Hyperparameter Control Method for a Network Intrusion Detection System Based on Proximal Policy Optimization", Published: 14 January 2022, MDPI.

[49] M. Zolotukhin, S. Kumar, and T. Hamalainen, "Reinforcement learning for attack mitigation in SDN-enabled networks," in Proceedings of the 2020 IEEE Conference on Network Softwarization: Bridging the Gap Between AI and Network Softwarization, NetSoft 2020, 2020, pp. 282–286.

[50] Jianfeng Yang et al.," Federated AI-Enabled In-Vehicle Network Intrusion Detection for Internet of Vehicles", MDPI, 9 November 2022, Electronics 2022, 11, 3658. https://doi.org/10.3390/ electronics11223658

[51] Ines Ortega-Fernandez, Francesco Liberati," A Review of Denial of Service Attack and Mitigation in the Smart Grid Using Reinforcement Learning", Energies 2023, 16(2), 635; https://doi.org/10.3390/en16020635, Jan 2023.

[52] Y. Liu, M. Dong, K. Ota, J. Li, and J. Wu, "Deep reinforcement learning based smart mitigation of DDoS flooding in software-defined networks," in 2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), pp. 1–6, IEEE, 2018.

[53] Wei, F.; Wan, Z.; He, H."Cyber-Attack Recovery Strategy for Smart Grid Based on Deep Reinforcement Learning", IEEE Trans. Smart Grid 2020, 11, 2476–2486.

[54] S. Kim et.al., "Deep Reinforcement Learning-Based Traffic Sampling for Multiple Traffic Analyzers on Software-Defined Networks," IEEE Access, vol. 9, pp. 47815–47827, 2021.

[55] Kamalakanta Sethi et al.," Attention-based multi-agent intrusion detection systems using reinforcement learning", Journal of Information Security and Applications, Elsevier, 2021, Volume 61, September 2021, 102923.

[56] Lei Zhang et al.," A Hidden Attack Sequences Detection Method Based on Dynamic Reward Deep Deterministic Policy Gradient", Security and Communication Networks, Volume 2022 | Article ID 1488344 | https://doi.org/10.1155/2022/1488344

[57] Chengming Hu et al.," Reinforcement Learning-Based Adaptive Feature Boosting for Smart Grid Intrusion Detection", IEEE Transactions on Smart Grid, IEEE, DOI: 10.1109/TSG.2022.3230730, 20 December 2022.

[58] Gang Luo, Zhiyuan Chen, Bayan Omar Mohammed, "A systematic literature review of intrusion detection systems in the cloud-based IoT environments", DOI: 10.1002/cpe.6822, 9 December 2021, Wiley research article.

[59] G. Y. Zou, "Toward using confidence intervals to compare correlations.," Psychological methods, vol. 12, no. 4, p. 399, 2007.

[60] David Silver et al.," Deterministic Policy Gradient Algorithms", Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

[61] CHUANLONG YIN et al.," A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks", IEEE Access, Digital Object Identifier 10.1109/ACCESS.2017.2762418, Nov 2017.

[62] Maya Hilda Lestari Louk et al.," PSO-Driven Feature Selection and Hybrid Ensemble for Network Anomaly Detection", Big Data Cogn. Comput. 2022, 6, 137. https://doi.org/10.3390/bdcc6040137.

[63] Thi-Thu-Huong Le et al.," Network Intrusion Detection Based on Novel Feature Selection Model and Various Recurrent Neural Networks", Applied Sciences, April 2019.

[64] Wenjie Zhang et al.," Wireless sensor network intrusion detection system based on MK-ELM", Springer, Soft Computing (2020) 24:12361–12374, https://doi.org/10.1007/s00500-020-04678-1(0123456789().,-volV)(0123456789(). ,- vol V), Jan 2020.

[65] Yanqing Yang et al.," Building an Effective Intrusion Detection System Using the Modified Density Peak Clustering Algorithm and Deep Belief Networks", Applied Sciences, 10 January 2019.

[66] Kamalakanta Sethi et al.," Attention-based multi-agent intrusion detection systems using reinforcement learning", Journal of Information Security and Applications 61 (2021) 102923.

[67] Manuel Lopez-Martin et al.," Application of deep reinforcement learning to intrusion detection for supervised problems", Expert Systems with Applications, Volume 141, 2020, 112963. https://doi.org/10.1016/j.eswa.2019.112963.

[68] JooHwa Lee et al.," GAN-based imbalanced data intrusion detection system", Personal and Ubiquitous Computing (2021) 25:121–128, Nov 2019.

[69] Kamalakanta Sethi et al.," A context-aware robust intrusion detection system: a reinforcement learning-based approach", International Journal of Information Security https://doi.org/10.1007/s10207-019-00482-7, Dec 2019.

[70] Tran Hoang Hai et al.,"NETWORK ANOMALY DETECTION BASED ON LATE FUSION OF SEVERAL MACHINE LEARNING ALGORITHMS", International Journal of Computer Networks & Communications (IJCNC) Vol.12, No.6, November 2020 DOI: 10.5121/ijcnc.2020.12608.