ACTOR CRITIC APPROACH BASED ANOMALY DETECTION FOR EDGE COMPUTING ENVIRONMENTS

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

[1]Department of Electronics & Communication, BNMIT, Bangalore, Karnataka, INDIA
[2]Research Head, New Horizon College of Engineering, Bangalore, Karnataka, INDIA.

ABSTRACT

The pivotal role of data security in mobile edge-computing environments forms the foundation for the proposed work. Anomalies and outliers in the sensory data due to network attacks will be a prominent concern in real time. Sensor samples will be considered from a set of sensors at a particular time instant as far as the confidence level on the decision remains on par with the desired value. A "true" on the hypothesis test eventually means that the sensor has shown signs of anomaly or abnormality and samples have to be immediately ceased from being retrieved from the sensor. A deep learning Actor-Criticbased Reinforcement algorithm proposed will be able to detect anomalies in the form of binary indicators and hence decide when to withdraw from receiving further samples from specific sensors. The posterior trust value influences the value of the confidence interval and hence the probability of anomaly detection. The paper exercises a single-tailed normal function to determine the range of the posterior trust metric. The decision taken by the prediction model will be able to detect anomalies with a good percentage of anomaly detection accuracy.

KEYWORDS

Reinforcement Learning, Actor Critic, Security, Anomaly Detection, Posterior Belief

1. INTRODUCTION

Information and computing are ubiquitous in the sphere of communication. One may refer to information being available and handled at the user devices while the other may refer to the information at the core cloud infrastructure. Information also gushes its way through various intermediate communication networks and servers. Challenges of IoT (Internet of Things) based networks like real-time massive data generation, heterogeneous data, dynamic demeanor networks, constrained memory, and resources are still difficult to elucidate. However, the commendation goes to researchers who have shed light on how to effectively glean embedded intelligence features from Machine Learning (ML), and Deep Learning (DL) techniques to incorporate them in IoT devices and networks.

There will an explosive growth in computationally expensive sensor data due to the erratic rise in sensory devices like wearable devices, smart phones, daily use appliances, vehicles, etc., Data analysis and computing will be highly challenging. Captured data will most of the time be of high dimension. Network handling capacity and handling delay mitigation requirements will pose stringent challenges too. Real-time inferences and responses are critical in a majority of applications. With more and more data being generated from the physical world, there is a definite boost in bottlenecks concerning network bandwidth and transmission speed too.

The authors of [1] have consolidated the above challenges into mainly latency, scalability, and privacy. Users send data to the cloud and this bears privacy concerns. Exposure to network attacks is always at the higherside of probability when it comes to data handling. Anomaly detection refers to the identification of an anomalous or abnormal activity in the EC network predominantly due to an attack. Security and privacy assurance should be mitigated and handled effectively in each every network zone/layer in all ”edge computing” system designs. The authors of [2] clearly categorize the security challenges in the 4layers namely (a) core cloud (b) edge servers (c) edge networks and (d) edge devices. The various attacks can include spoofing, Man-in-the-Middle, Rogue, Denial of Service (DoS), Distributed Denial of Service (DDoS) and other smart network attacks influencing the performance. However, our work concentrates on detection ofanomalies and outliers with respect to attack detection irrespective of the type of attack. A sustainable, rewarding edge computing model should abide by (a) Data Confidentiality, (b) Data Integrity, (c) Authentication, (d) Access Control and (e) Privacy and Security.

Our work will mainly streamline the security of data based on anomaly detection. The work explores the possible avenues of using cutting-edge deep learning algorithms in the security domain, extensively for edge computing environments. Henceforth, the discussion will be concerning the following four dimensions namely (i) Anomaly Detection (ii) Network Security (iii) Edge Computing Environments (iv) Deep Learning based approaches and Algorithms.

1.1. Convergence of Edge Computing and Deep Learning

We now understand that a centralized, traditional cloud computing model is not very encouraging. Edge Computing (EC) promises to be a viable alternative or can say a probable solution to most of the network challenges. Edge Computing is a novel architecture wherein the services of the cloud are protracted up to the network edge. This reduces latency. The platform is maintained very close to the data source, this supports effective, speedy real-time processing, data optimization, security, and privacy.[3]. The authors of [4] have highlighted the advantages of edge computing as low latency, energy saving, context awareness, privacy, and security. Scale reduction in edge computing makes these networks less prone to attacks as compared to the large-scale data centers of the cloud.

The image in Fig. 1 represents the exposure of edge computing architecture to different network attacks. Various distributed, dispersed, appropriated low latency and dependable astute services alongside DL are ensured by the edge networks. The computational efficiency and edge user experience for time-critical applications improve and localized user experience could be improved significantly with edge computing when they discuss cloud offloading.

Deep Learning (DL) is a catalyst for edge intelligence DL is a machine learning technique that has originated from Artificial Neural Networks (ANN).

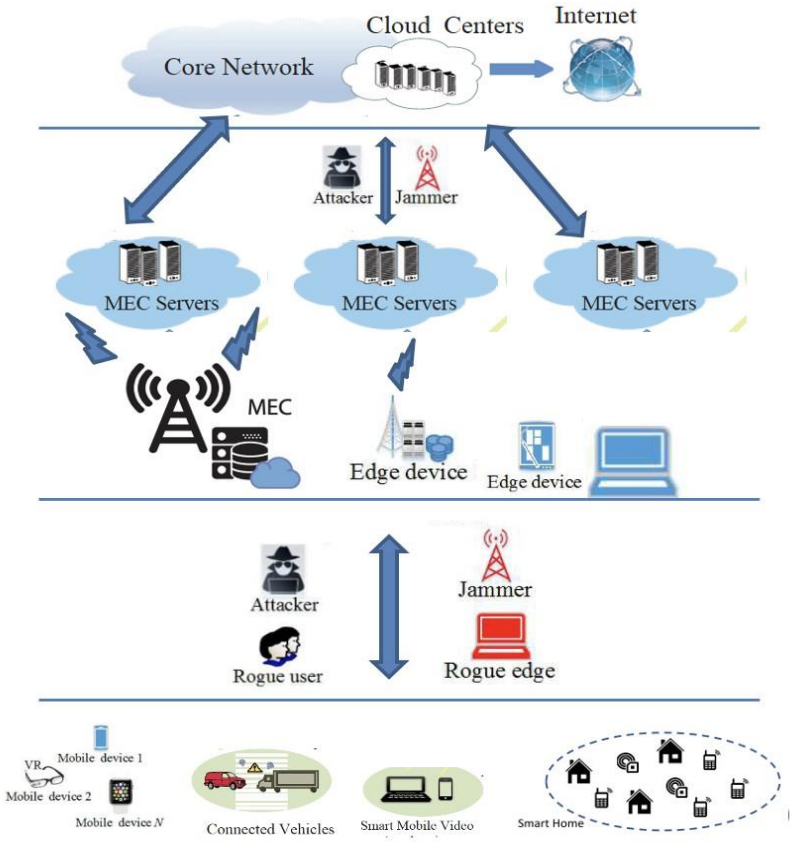

Figure 1. Edge Computing Environment subject to attacks

The ”DL - edge computing” combo is guile but in-depth understanding and interpretation of DL models, edge computing features and also design, development and deployment standards is a major requirement. DL is known for distributed computing and, analysis of unlabelled, uncategorized, and unsupervised data [5].There are many real-time complex machine learning tasks which utilize deep learning have demonstrated good performance, heeding to privacy and latency concerns.

The category of reinforcement learning (RL) algorithms works towards securing data transfer at the edge environments. RL works on the principle of ”reward function” system. The agent learns from the feedback post its interaction with the environment. Agent’s action will be evaluated based on the reward. This kind of a DL technique is highly suggested for dynamic edge computing environments wherein learnings are based on experiences rather than on the data.
A RL environment is usually cast as a Markovian Decision Process (MDP) which synthesizes the entire concept based on a generic computational framework which will be used by the model to draw conclusions or otherwise decisions.

The authors of [6] summarize few compelling RL based security methods which promise to better the security parameter in the ”edge”. However, RL-based edge security solutions still welcome a lot of challenges which are to be addressed.

Q-learning is a prominent RL technique which has proved its competency in spoofing, malware, jamming and eavesdropping management. Q-values revolving around the learning rate and discount factor are updated by the iterative Bellman equation. Learning performance witnesses an

uplift for stochastic environments. Unfortunately, edge security solutions implemented via Q-learning face high-ambit issues. The ”model-less”, online learning method has no advance information about the working environment unlike ”Markovian”. Q-learning variants like the ”Dyna Q” offer to be a savior then. Finally, the conclusion statement that can be put across is that both MDP and Qlearning project higher culminations as against convex optimization. Also, consummation of Q-learning will be unstable for lesser number of learning times. [7].

The authors of [8] mention about security defense mechanism in Deep Neural Network based systems. However, the work tries to focus on providing an authentication strategy based on Reinforcement learning. The proposed work focuses on the Actor Critic (AC) algorithm to detect anomalies in the network. DNNs proximate the state-action duo’s value function. The critic associated gradient information is available for analysis. It is observed that especially for real-time random processes and continuous domain environments, model approaches based on policies and protocols would be a good bet. Even though value-based approaches are stable and sample efficient, the former will have a very impressive, faster convergence rate. As far as the ”state-action duo” values are continuous space entities in seemingly stationary environments. A simple working idea of Actor-Critic is depicted in the Fig. 2 below.

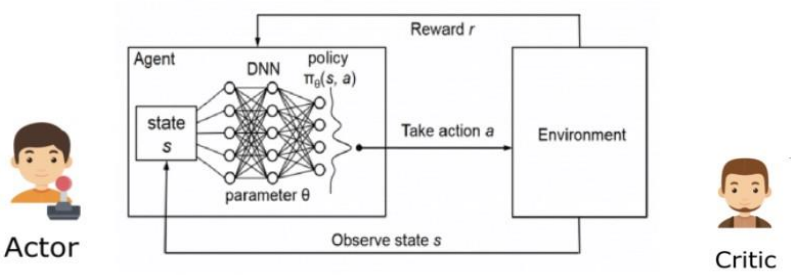

Figure 2. Actor-Critic environment.

1.2. Network Architecture for MEC Security

A better option in terms of latency and security is to offload data at the resource efficient edge servers which are well equipped to execute the DNN models that are provided. The deliverable capabilities of deep learning algorithms can be experienced if the challenges with respect to edge devices and the edge environment as a whole are made to move towards efficient solutions. The authors of [9] provide an insight into few of the challenges in the era of edge computing. The choice of a deep neural network, model partition and allocation scheme all have a role to play in security and privacy preservation.

A specialized DNN suited for temporal data is the Recurrent Neural Network (RNN)which has loops in their layer connections so as to store the state value and envision the sequential inputs [10].

The prior computed values influence the output. The “memory” catches in RNNs resorts to confiscate the information calculated so far. RNN architecture supports data persistence and models short term dependencies in a fine-tuned manner. The shortcomings of RNN include the hardship in training, the exploding-vanishing gradients during training, inability to handle very long sequences in the case of tanh and relu used as activation functions. Long-term dependencies will also impact good results.

The ”Vanishing Gradient” problem seeks a good solution from Long Short-Term Memory (LSTM) networks. The gradient can pick up patterns over larger regions of the sequence. LSTM networks have memory blocks that are connected into layers instead of neurons. The Forget, Input and Output gates constitute the total working of the LSTM model. The Forget gate discovers what details are to be discarded from the block. The Forget gate reviews at the previous state and the input, delivers a value well lesser than 1 for each number in the cell state.

The input gate identifies which value from input should be used to modify the memory/ store in the cell state. First, a sigmoid layer called the “input gate layer” decides which values to update. Next, a tenth layer creates a vector of new candidate values, that could be added to the state. In the next step, these two values will be combined to create an update to the state.

The role of the sigmoid layer is to check and deliver selected parts of the cell state to the output end. The cell state is cascaded with ”tenth”, following which the gate output is multiplied with the cascaded factor.

1.3. Contributions

The contribution of the proposed work is oriented towards providing a multiple layer-based implementation of a security model. The criticalities of security in communication networks are analysed via detection of anomalies with the aid of a powerful deep learning algorithm - The Actor Critic algorithm.

1) The model uses of LSTM based Recurrent Neural Networks as against Convolutional Neural Networks for active hypothesis testing to capture the temporal dynamics of the sensory environment. The attack detection model is a binary indicator, based on which the system decides when to stop receiving sensor data from a specific sensor.
2) Actor Critic algorithm is implemented with both actor and critic taken as separate networks which interact with each other to improvise on the detection accuracy of network anomalies. The work banks upon the” criticise-to-improve” working principle of the algorithm.

1.4. Organization of the Paper

Section III briefly explains the proposed system model which includes the problem formulation, and establishment of the hypothesis. The section also discusses how to solve the active hypothesis testing problem and the selection of posterior belief thresholds. Section IV reveals the deep actor-critic framework for optimal policy selection.MDP Parameterization, modeling equations of both Actor and Critic framework, and reward computation are highlighted in the section. Section V gives an outline of the algorithm. Details of the confusion matrix are explained in Section VI. Section VII deals with the results and evaluation. The conclusion and opportunity for future work are enclosed in Section VIII.

2. METHODOLOGY

2.1. Anomaly Detection Environment

The correlation between features is one of the main reasons for choosing the LSTM neural network. Recurrent neural networks are a wise option to handle use cases of hypothesis testing which will be discussed elaborately in the upcoming sections. Traditional time-series models analyze time-series as separate entities and do not consider the complex inter dependencies among the different timeseries [11]. Also, the Q-function is approximated by LSTM in order to

tackle the attacking agents due to its potential to capture the temporal dynamics of the environment [12].The proposed model will determine whether an anomaly has been detected in the input. Eventually, output of the attack detection model should be a binary value that indicates '1' in case of attack and '0' in case of no attack. Input database can be split as training and testing data sets. The core architecture includes the three vital layers of LSTM. A suitable activation function and loss function is chosen in the FC layer. Dropout will be used to mitigate overfitting issues of the anomaly model. The model can be improved using an apt optimizer as well. The closure of the model is taken care by a single neuron output layer which indicates the status of attack. The Fig. 3 below represents the feasible LSTM architecture for anomaly or attack detection.

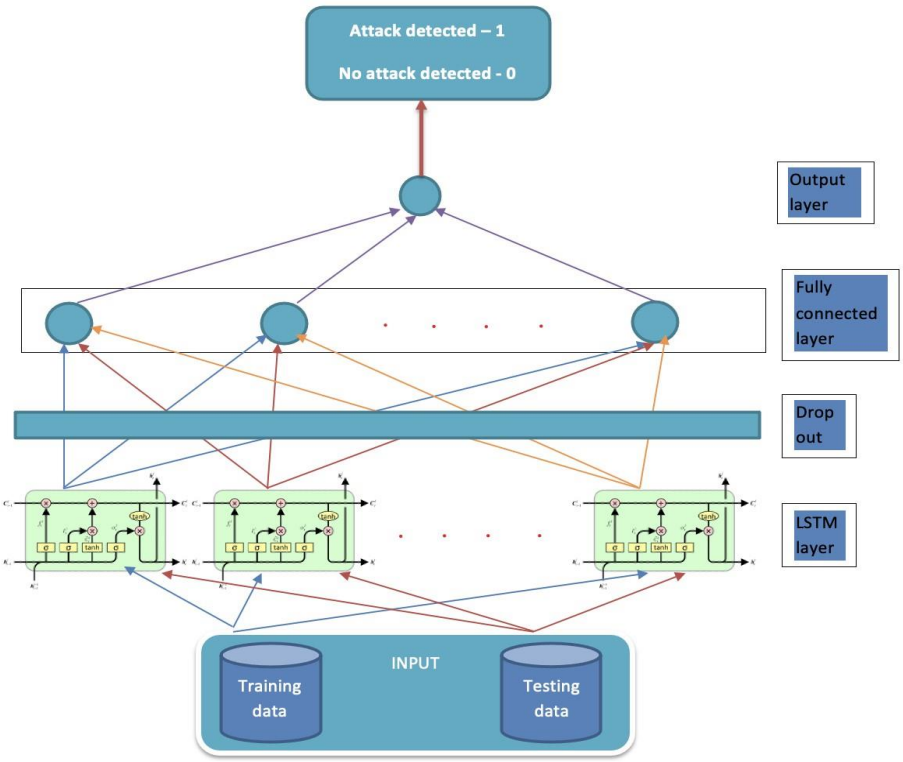

Fig. 3. LSTM architecture for anomaly or attack detection.

The proposed work considers the fact that edge data will be fetched from various sources through sensors in real-time. Sensors are supposed to provide impeccable data to the receiver end. Sensors are many a time less reliable in content delivery either due to sensor construction, environmental changes, lack of sensor calibration. However, our discussion is confined to non-reliability of sensor data due to network attacks only.

The model proposed aims at a trust/ satisfaction metric which will vary proportionally with the user provided QoS which indirectly is a measure of successful detection accuracy. A potent reinforcement algorithm, the Actor-Critic is used to dynamically help the agent stretch out towards a maximized reward. The agent's main target will be to maximize the average value of the user response in terms of the satisfaction metric (=), which is part of the reward calculation. Each time the agent performs an action on the environment, the critic evaluates/criticises the actor's performance. The fundamental goals of a lucrative agent can be the following: (1) Maximize the average reward function/ trust metric (2) Optimizing latency with respect to

offloading in MEC environments (3) Reducing the stopping time for a quick turnaround on the sensed data [13]. However, the work focuses on only maximizing the trust metric so as to bring in the security feature into the model.

Considering each independent sensor source as an independent process, we will look forward to a model which can handle (P) processes. Inherently, the word "process" refers to the "sensor". It is obvious that each process is random and the probability of (say p) processes out of these (N) processes being anomalous/ atypical (ρ_{ad}) is acceptable. At a given time instant (t), (N) can take values in the range [0,P].The assumption is that (N) does not exceed the value of (P) at any point in time. The decision taken by the prediction model must be able to detect the anomaly with an appreciable percentage of anomaly detection accuracy.

There will be 2^P values that can fit into the state vector say (V) for (P) processes/ sensors. The paper brings in the concept of state and action since they are required to be used in the Actor-Critic implementation. The vector states can be represented by sample space $V \epsilon (0,1)^P$.The algorithm will cater to all available sensors hence facilitating $2^P - 1$ actions [14].

2.2. Problem Formulation– The System Model

Trust metric and confidence interval/satisfaction metric are the deciding factors for maximization of reward in the proposed work. The satisfaction metric is the Bayesian log-likelihood ratio of hypothesis at time (t) given as:

$$\mathfrak{Jq}\ (\boldsymbol{\beta}) = \log \frac{\boldsymbol{\beta(q)}}{\boldsymbol{1-\beta(q)}} \quad \textbf{(1)}$$

Here, β_q is the qth entry of a posterior belief vector β.

The average Bayesian log likelihood ratio contributes to the reward component. Rewards can be averaged as:

$$\boldsymbol{R\mu(t) := \frac{1}{\tau}\sum_{t=1}^{\tau-1} E^{\mu}\,[\,r_{\mu}\,(t)]} \quad \textbf{(2)}$$

where $r_\mu(t)$ is the instantaneous reward of MDP given by:

$$\boldsymbol{r_{\mu}\,(t) := \mathfrak{J}_{avg}\,(\beta(t))\ -\ \mathfrak{J}_{avg}\,(\beta(t-1))} \quad \textbf{(3)}$$

Posterior belief threshold influences the probability of anomalous behaviour which is defined with the help of a single-tailed function. The hypothesis tests true when anomaly is detected and anomaly is detected when posterior trust value is within the range say (α,β_{max}) which define the lower and upper bound of posterior values.

$$\boldsymbol{\rho_{ad}\,(\beta)} = \begin{cases} \boldsymbol{F(\alpha)} & \boldsymbol{\beta} < \alpha - hypothesis\ failure \\ \boldsymbol{F(\beta_{max})} & \boldsymbol{\alpha \leq\ \beta < \beta_{max} - hypothesis\ success} \end{cases} \quad \textbf{(4)}$$

The aim of the work in terms of the algorithm is to maximize the instantaneous reward of the MDP $r_\mu(t)$ depending on the average Bayesian log likelihood ratio $=_{avg}(\beta)$. The algorithm ensures to minimize the squared value of temporal error $\psi_{\mu\varphi}(t)$. The Actor update towards the target value follows.

2.3. Establishment of Hypothesis

Sensor data can be used to assess the believability and validity of a hypothesis. This is what is referred to as "Hypothesis testing." Herman Chernoff was one of the first to formulate active hypothesis testing [15]. The model has to administer to infer the hypothesis for a given set of observations and learn the optimal selection policy. At time (t), we define a sample space ζ (t) which shall constitute all samples that are captured from different sensors at that time instant (t).Let us denote the set by say $\{O_1,O_2,.....O_P\}$.Therefore, the corresponding observations of each sensor at time (t) can be represented as $O_{a(t)} \in \{0,1\}$where a(t) refers to the process observed at each sensor given as a(t) $\in \{1,2,3,....P\}$.We can refer to the sample from a particular sensor or a particular process (i) as well.

The active hypothesis testing problem equivalent to the anomaly detection problem has 2^P hypothesis. Out of all the unknown number of processes, a hypothesis test will be defined as $H_q$for all values varying from q = 1,2,.....2^P. It is to be noted that out of all samples on the observation space, samples will be collected from the sensor as long as hypothesis $H_q$is verified to be logically true and the others when the hypothesis $H_q$is false. If at any time the hypothesis tests "true", that eventually means that the process has tested positive for anomaly or abnormality and samples have to be immediately stopped from being retrieved from sensor. The supply/sensor chain has to be terminated.

2.4. Establishment Solving the Active Hypothesis Testing Problem

Each of the possible states of the processes are associated with a hypothesis which is proposed by the decision maker. Later, the posterior probabilities are computed based on the hypothesis laid out. Using all the information, the agent forms a posterior trust value β(t) on hypothesis $H_q$at time (t). The observation information available with the agent at any time (t) is given by:

$$\boldsymbol{O_\tau} \subseteq [\boldsymbol{P}]_{\boldsymbol{1:t-1}} \quad \textbf{(5)}$$

Sequence of actions selected by the agent based on the critic's feedback can be represented as:

$$\boldsymbol{A_\tau} \subseteq [\boldsymbol{P}]_{\boldsymbol{1:t-1}} \quad \textbf{(6)}$$

The optimal policy design depends upon the estimation algorithm. One can foresee two different aspects of the belief vector. In the first case, the belief vector is the posterior probability that the j^{th}process is non-anomalous. Therefore, the trust vector can be updated on the arrival of each observation as the probability of the state being '0', as shown in the equation.

$$\boldsymbol{\beta_j(t) = \rho(s_j = 0 \mid O_\tau, A_\tau;\ \tau = 1,2,3,\ldots(t-1))} \quad \textbf{(7)}$$

In the second case, we denote $\beta_q(t)$ as the posterior trust value of the hypothesis $H_q$being true at time (t) such that $\beta_q(t) \in [0,1]^{2^P}$.

$$\boldsymbol{\beta_q(t) = \rho(H = q \mid O_\tau, A_\tau;\ \tau = 1,2,3,\ldots(t-1))} \quad \textbf{(8)}$$

Bayes rule administers a way to positively refurbish the beliefs based on the arrival of fresh samples of evidence. In our case, we were trying to compute the probability value if and only if the given hypothesis is considered true. Given additional evidence such as the sequence of actions, we can update our probability. Estimations can be improved based on the effective usage of the available prior knowledge. The general final form of Bayes rule is formulated as:

$$\boldsymbol{\rho}\ (\boldsymbol{A}|\boldsymbol{B}) = \frac{\boldsymbol{\rho}(\boldsymbol{A}).\boldsymbol{\rho}(\boldsymbol{B}|\boldsymbol{A})}{\sum_{q=1}^{H}\boldsymbol{\rho}(\boldsymbol{A_q}).\boldsymbol{\rho}(\boldsymbol{B}|\boldsymbol{A_q})} \quad \textbf{(9)}$$

The final form of Bayes rule can be applied to our study by replacing (A) with hypothesis and (B) with the corresponding observations.

$$\boldsymbol{\beta_q(t)} = \frac{\boldsymbol{\rho}(\boldsymbol{H=q}).\boldsymbol{\rho}(\boldsymbol{Z}[\boldsymbol{A}_{\boldsymbol{\tau}}(\boldsymbol{\tau})]|(\boldsymbol{H=q})}{\sum_{q=1}^{H}(\boldsymbol{H=q}).\boldsymbol{\rho}(\boldsymbol{Z}[\boldsymbol{A}_{\boldsymbol{\tau}}\ (\boldsymbol{\tau})]|(\boldsymbol{H=q})} \quad \textbf{(10)}$$

It is to be noted that τ varies up to the value of (t-1) starting from unity and $Z[A_{\tau}(\tau)]$ is the corresponding sensor measurement or the corresponding observation. Observation probabilities corresponding to one of the hypotheses are segregated by the decision maker, taking into account that the value exceeds the desired confidence level β_{high}.

Confidence intervals are widely accepted as a preferred way to present study results. Confidence intervals measure the uncertainty of an estimate. Maximum Likelihood Estimation (MLE) methods provide 2 major approaches to construct the confidence interval. The first being the ”Asymptotic” method wherein the likelihood function can be approximated by a parabola around the estimated parameters. Assumption is its validity for very large samples; otherwise, the performance of the confidence interval will be poor, considerably less than the nominal rate of 95%. Confidence limits (β_{low}, β_{high}) reflecting the asymmetric sampling distribution of data may be based on different variance estimates [16]. The second approach is the Likelihood Ratio Test(LRT), an experiment performed by R.A.Fisher to find the best overall parameters value and likelihood. LRT is range preserving and behaves in a predictable manner as the sample size grows.

The confidence interval has two unusual features: (i) The endpoints can stray outside the parameter space; that is, one can either get β_{low} <0 or β_{high} >1. (ii) The confidence interval width becomes zero if no successes or failures are observed.

A possible solution to ensure that the endpoints do not cross over the parameter space is to parameterize using a completely different scale like”log-odds/ logit/ logistic transformation”. The logit function is the inverse of the logistic sigmoid function $1/(1+e^{-x})$. It is extensively used in statistics and machine learning to quantify confidence level [17].

The intent is to evolve with a compelling selection strategy or an optimal policy which the actor/agent can implement and strive towards an increased value of confidence level = on the hypothesis (H) being true. In response to all the sensor measurements, one can perform the reward calculation. We define the average Bayesian log likelihood ratio as:

$$\boldsymbol{\Im_{avg}(\beta)}\ =\ \sum_{q=1}^{H}\boldsymbol{\Im_q(\beta).\beta_q} \quad \textbf{(11)}$$

Asymptotic expected reward is based on the average rate of increase in the confidence level on the true hypothesis H and is defined as [18]:

$$\mathbf{R}(\boldsymbol{\mu}) := \lim_{\boldsymbol{O_\tau}\to\infty}\frac{\mathbf{1}}{\boldsymbol{O_\tau}}\boldsymbol{E^{\mu}}\ [\boldsymbol{\Im(\beta(O_\tau\ +\ 1)\ -\ \Im(\beta(1)]} \quad \textbf{(12)}$$

However, the proposed work extends the concept by averaging over all the measurements obtained at different time intervals for a particular sensor. Maximizing the reward can be devised as an infinite horizon, average-cost Markovian Decision Process (MDP) problem. The posterior trust vector $\beta(t)$ will act as the state of the MDP. Henceforth, the average of rewards can be formulated as in (2).

2.5. Selection of Posterior Belief Thresholds

It has been highlighted earlier those sensory measurements from various sensors are stochastic and random in nature. Henceforth, it would be highly likely to define the probability (ρ_{ad}) of anomalous/ atypical behavior in the data indicating the influence of a network attack.

For an anomaly detection task, the posterior trust value (β) influences the value of the confidence interval and hence even the probability of attack/ anomaly detection. As (β) increases, (ρ_{ad}) also escalates. According to the above analysis, we employ a singletailed normal function, $\rho_{ad}(\beta)$, to reflect the relationship between the vehicle speed and the task delay constraint. The properties of the one-tailed normal function are shown in Fig.4.

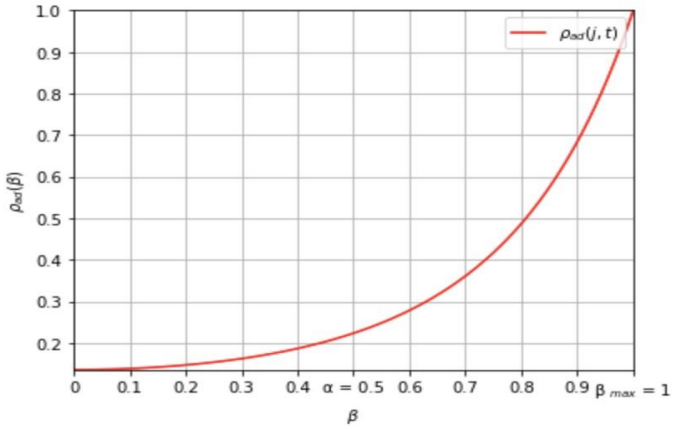

Fig. 4. Posterior Trust and Anomaly probability modeling for Anomaly detection.

When the posterior trust value is within the range of (0,α), the function $\rho_{ad}(\beta)$ is well within the boundary of the hypothesis being rejected i.e; anomaly has not been detected. On the contrary, When the posterior trust value is within the range of (α,β_{max}), the hypothesis is in acceptable condition indicating that an anomaly has been detected. The algorithm has to now reach the ”termination” condition. It is suitable to use the following function to describe the Posterior Trust and Anomaly probability model for anomaly detection as follows:

$$\boldsymbol{\rho_{ad}\,(j,t)\;=\;exp\,(\frac{\beta_{j,t}^2-\beta_{max}^2}{2\alpha^2})} \qquad \textbf{(13)}$$

$\beta_{max} \approx 1$ and in order to ensure that the probability that anomaly detection probability is within the maximum value exceeds 95%, we denote the following [19]:

$$\boldsymbol{\alpha\;=\;\frac{\beta_{max}}{1.96}} \qquad \textbf{(14)}$$

3. Deep Actor-Critic Framework for Optimal Policy Selection

This section describes the learning framework to solve the anomaly detection problem. The frame of reference will bank on the advantages of a constructive-feedback oriented feedback critic network, an advantage function and TD learning-based update. All these highlighted features are promoted by the Actor-Critic framework wherein a trained critic model approximates the value function. The ”advantage” function conveys to the agent about the quantum improvement

required in comparison with the average action value taken at that state. Alternately, this function computes the extra reward that the agent gets if taking that action.

The Actor and Critic networks are discussed in detail in the subsections. The deep reinforcement learning based algorithm will emphasize on how to interact with the environment effectively and how to learn effectively from experiences.

3.1. MDP Parameterization

The underlying fundamental principle of the Actor-Critic is the MDP which is characterized by four fundamental segments namely:

1) Time-indexed $STATE$ variables S_t in finite state space (S) The posterior trust vector β_t will be the state of the MDP at time (t). We consider the time index since the β value changes dynamically with each sensor measurement arriving at different time instants.
2) Time-indexed $DECISION$ or $ACTION$ variables a_t in finite action space (A) - The agent takes the decision, an action (a) with the help of a constructive critic and transitions to a new state $\beta(s,a)$.
3) The optimal policy in general has to maximize the average REWARD value. Reward calculation helps to evaluate the agent's action on the environment. Specifically, in the work considered, policy has to augment and escalate the averaged value of confidence level metric =. This ensures that the decision accuracy is definitely boosted. maximize the average confidence level metric = and hence maximize the decision accuracy.

3.2. Actor Framework

The Actor and Critic networks will be two separate neural networks which interact with each other productively. The actor explores and learns the policy (μ) on the true hypothesis characterized by say (φ) which maps the posterior trust distribution (β)to the action space i.e; chooses a valid action based on the posterior probabilities (β).Valid action refers to selecting the corresponding sensor and receiving the sample to update the posterior belief. In each iteration, the agent will score all valid actions, and choose the one with the highest score to execute.

The actor network's output layer provides an assemblage of all actions that belong to the action space; the elements in this action space can be treated as the subset of [P].

$$\boldsymbol{a_t} \in \boldsymbol{A} := \boldsymbol{\mu_\varphi}\,(\boldsymbol{\beta(t-1)}) \qquad \textbf{(15)}$$

3.3. Critic Framework

The critic portrays its network as a "state-value function approximator". In fact, the actor updates its policy based on this value function computed by the critic.

The new state is subject critic-based computation and this happens after each action selection. Results obtained can either be better than the prior or even worse. Critic receives the following data as input:

1) Posterior trust value at time (t) - $\beta(t)$
2) Instantaneous Reward value $r_\mu(t)$ from equation [11]
3) Posterior trust value at time $(t-1)$ - $\beta(t-1)$

TD error can be used to evaluate the action implemented (a_t) for the particular state (S_t). If the TD error is positive, it suggests that the inclination to (a_t) must be intensified for the upcoming observations. On the contrary, for negative TD error, inclination must be weakened for future values.

3.4. Bringing in Reward Computation

Critic is responsible for temporal error calculation and can be expressed mathematically as:

$$\boldsymbol{\psi_{\mu\varphi}(t) = r_{\mu}(t) + \lambda.V(\beta(t)) - V(\beta(t-1))} \quad \textbf{(16)}$$

In the above equation, (V) is the estimate of the current value function and (λ) is the discount factor (0,1). Discount factors become important to be introduced especially in case of time-invariant samples or stationary values. It is required to obtain an optimal value function to determine the best fit line to data by using Mean Squared error (MSE). The critic updates itself by minimizing the squared value of temporal error (ψ^2).

3.5. Updating the Actor and it's Parameters

Policy gradient helps the actor to get updated. TD error (ψ) from equation [16] will be used to compute the gradient as depicted in the equation below:

$$\boldsymbol{\nabla_{\varphi}.J(\varphi) = E_{\mu\varphi}\,[\nabla_{\varphi}.log\,\mu_{\varphi}(\beta(t-1),a_t).\psi^{\mu\varphi}\,]} \quad \textbf{(17)}$$

In the process of TD learning, the weights will get updated at each learning step in real-time. The technique does not wait till the end of the episode to complete update of the weights. This supreme characteristic overcomes the shortcomings of the policy gradients. $\mu_{\varphi}(\beta(t-1),a_t)$ refers to the score for the selected policy i.e; how well did the actor faired with the current policy, it's a sort of evaluation. $\nabla\varphi$ represents the gradient with respect to the parameter φ.

The actor also is responsible for updating φ by taking the support of the policy gradient and TD error value.

$$\boldsymbol{\varphi_t = \varphi_{t-1} + \psi_{t-1}.\nabla_{\varphi_{t-1}}\,[log\,V_{\varphi}(A_t|\,\beta(t-1))]} \quad \textbf{(18)}$$

Now, the paper denotes the weighted difference of actor parameters where η is the learning rate with values in the range set (0,1).

$$\boldsymbol{\eta.[\nabla_{\varphi}.log\,\mu_{\varphi}\,(\beta(t-1),a_t).\psi^{\mu\varphi}]} \quad \textbf{(19)}$$

Actor can now accept the criticism (positive/ negative) which has been provided by the critic network, update itself as below and work towards the target value to reach the optimal goal.

$$\boldsymbol{\varphi_t = \varphi_{t-1} + \eta.\left[\nabla_{\varphi}.log\mu_{\varphi}(\beta(t-1),a_t).\psi^{\mu\varphi}\right]} \quad \textbf{(20)}$$

The dual networks will be handled and trained solely by disembarking on the gradient value. This definitely ensures traversing in the direction in which value function is accelerated [20] (to find the global maximum) to update both their weights.

4. ALGORITHM OVER VIEW

Algorithm 1 Actor-Critic based anomaly detection

Parameters Used:

a. Upper limit value of number of episodes - $E_p = E_{p(high)}$
b. Discount Rate - $\lambda \, \epsilon(0, 1)$
c. Learning Rate - $\eta \, \epsilon(0, 1)$
d. Posterior Trust Range - $\beta \, \epsilon(\alpha, \beta_{max}) \equiv (0.51, 1)$ using [14]
e. Time - t
f. Samples are captured from different sensors at time instant (t) - $\zeta(t)$

5. INTERPRETATION OF THE CONFUSION MATRIX

The confusion matrix can be used effectively in machine learning to visualize vital predictive analytical parameters like accuracy, F1 score, recall score and precision score. Analysis of True Negatives (TN), True Positives (TP), False Positives (FP) and False Negatives (FN) help to measure the performance of the anomaly detection system.

$$\textbf{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN}$$

$$\textbf{Precision} = \frac{TP}{TP+FP}$$

$$\textbf{Recall} = \frac{TP}{TP+FN}$$

$$\textbf{F1} = \frac{Precision * Recall * 2}{Precision + Recall}$$

Preliminary Initialization:

Critic Network $V_\phi(O)$ Initialization with random weights
Actor Network $\mu_\phi(O)$ Initialization with random weights

Core Steps:

for E_p = 1 to $E_{p(}high)$ **do**
  Set $t = 0$
  Generate hypothesis H to be true according to β, making it the condition in the "while" loop
  while $\alpha \leq \beta < \beta_{max}$ **do**
    (1):Actor selects one out of the (P) sensors/processes according to the decision policy, chooses action as in [15] $a_t \epsilon A := \mu_\phi(\beta(t-1))$
    (2):Collect sensor measurements O_{a_t}, t
    (3):Calculate posterior trust value at time (t) as $\beta(t)$ using equation [6] based on values received in (2).
    (4):Agent obtains reward as in equation [11].
    (5):Critic network computes the TD error with the help of the new state value received [16].
    (6):Critic network is updated by minimizing the TD error with respect to $V(\beta(t)$
    (7):Update actor network as in [20].
    (8):Increment time index as $t = t + 1$
  end while
  Finalize the status of hypothesis
  Accept hypothesis (1)
  Reject hypothesis (0)
end for

6. RESULTS AND DISCUSSION

6.1. Selection of Dataset

Malicious input detection systems are trained on internet traffic record data. This research work makes use of NSL-KDD which is the most prevalent data set. The work has taken nearly 1,40,000 sample points maintaining a workable ratio of training and testing data as approximately 82% and 18% respectively. The subset datasets used are KDDTrain+ and KDDTest+ .The dataset samples have a combination of normal, Denial-of-service, Probe, User-to-Root and Remote-to-Local attacks. The data processing logic obtains n-row batch from the dataset and returns dataframe with correct labels for detection.

6.2. Selection of Hyperparameters

RMS Prop optimizer is used to accelerate the optimization process. The proposed work uses ReLu as the activation function. Table II shows a list of hyperparameters used and their respective values.

6.3. Simulation Results

Fig.5 represents the average value of accuracy of detection of network attacks. The bar plot gives a comparative visualization of correct estimates, false positives and false negatives. False negatives refer to those sensor samples which are attacked and yet are left undetected; this is a serious threat to the system performance. However, a marginally better detection accuracy is witnessed for the normal, DoS and Probe attacks.

Table I : hyperparameters with values

Hyperparameters	**Values**
Batch size	100
Activation function	ReLu
Optimizer	RMSProp
Loss Function	MSE
Discount factor	0.99
Learning Rate	0.00025

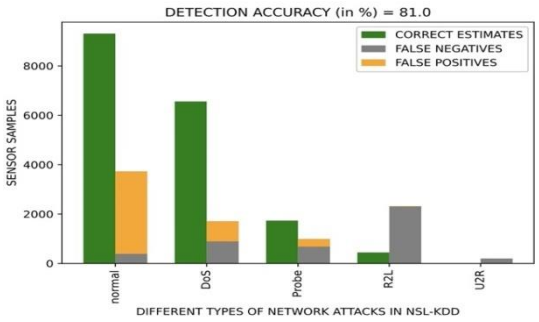

Fig. 5. Detection accuracy of Network Attacks in NSL-KDD

Fig.6 depicts the colour encoded heat map to represent the confusion matrix. Best True Positive value achieved for normal attacks according to the chart is nearly 0.96 .

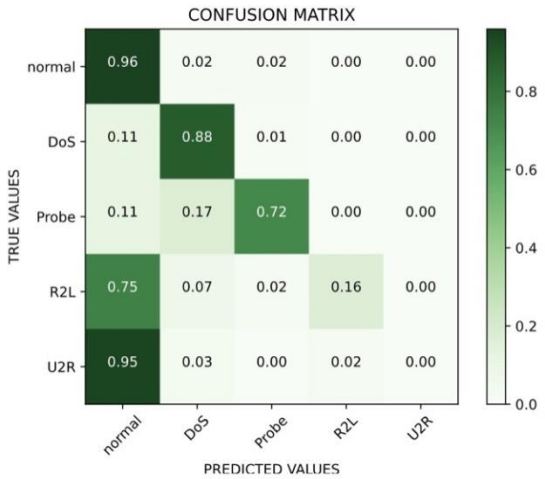

Fig. 6. Confusion Matrix of True versus Predicted values of attack detection.

Fig.7 depicts the colour encoded heat map to represent the action probabilities with respect to the states. The policy prediction function uses the Bayesian probability values and computes the action probabilities.

Fig.8 depicts the graph to represent the Bayesian Posterior probabilities which have been computed using the relation mentioned below.

$$\mathbf{P}(\boldsymbol{\varphi}|\boldsymbol{y}) = \boldsymbol{dist.pdf}(\boldsymbol{x}, \boldsymbol{aprior} + \boldsymbol{data}, \boldsymbol{bprior} + \mathbf{N} - \mathbf{data}) \quad \mathbf{(21)}$$

The Bayes theorem is used, priori probability is updated and then the value of posteriori probability is obtained. PDF of the posterior distribution must be the Probability Density Function (PDF) of the Beta distribution. Proportionality constant must be whatever constant is required to make this PDF integrate to 1 over p (0, 1).

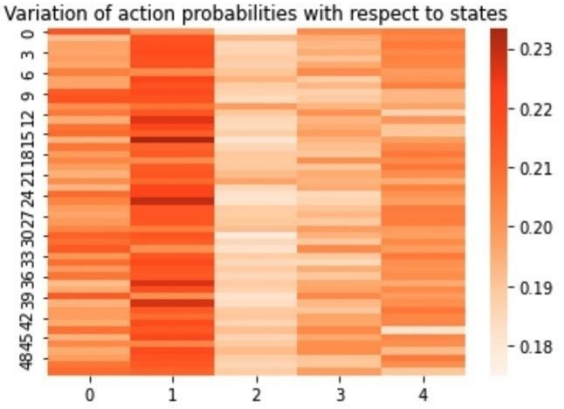

Fig. 7. Variation of action probabilities.

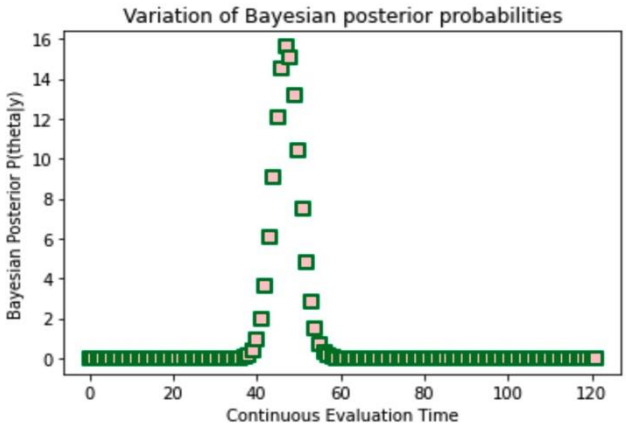

Fig. 8. Variation of Bayesian probabilities.

Fig.9 depicts the representation which demonstrates the variation of rewards and accuracy with respect to varying episode lengths. Rewards and accuracy values are highlighted separately in the graph. Bayes theorem will be used to establish independence between the system inputs and model parameters.

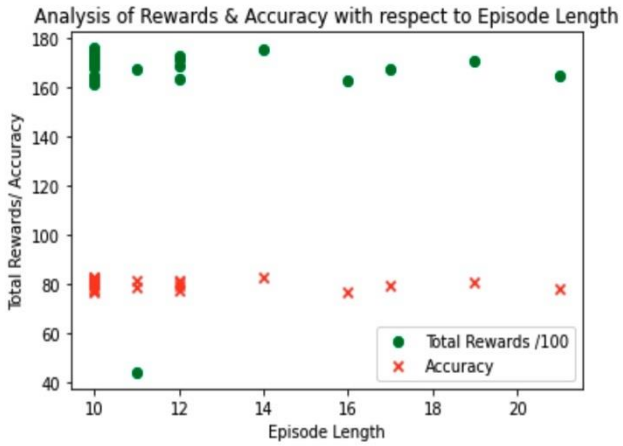

Fig. 9. Reward Analysis with respect to Episode Length

6.4. Discussion of Results

1) The analysis of Correct Estimates with respect to false negatives and false positives has shown a successful detection accuracy in case of Normal, DoS and Probe attacks.
2) Metrics of TP, FN and FP have been obtained and tabulated below for Normal, DoS and Probe attacks.

TABLE II VALUES OF VITAL METRICS

Name of Metric	TP	FN	FP	TN
Normal	0.96	0.04	1.92	2.17
DoS	`0.88	0.12	0.29	3.83
Probe	0.72	0.28	0.05	3.1

7. CONCLUSION AND FUTURE WORK

The proposed work implements the Actor Critic algorithm for anomaly detection. The deep reinforcement learning approach incorporates a LSTM network based Actor Critic algorithm for efficient anomaly detection. The algorithm presented aims to identify and fix the incongruous processes among a collection of binary processes. The analysis picks a single process at a time. The samples from sensors are further ceased from being received once anomaly has been detected.

The satisfaction metric which is the Bayesian log likelihood ratio of hypothesis is the determining factor for reward maximization. The hypothesis tests true on anomalous detection when posterior trust value is within the range defined by the bounding posterior values. Computations are based on marginal probabilities of the processes. The one-tailed normal function used in the proposed model helps to decide the range of the posterior trust metric. The range influences the decision of the hypothesis being rejected or accepted.

The scope for future work aims at minimizing the values of false negatives so as accelerate the detection accuracy. An optimal process selection policy can be proposed with an improvised deep reinforcement learning based Asynchronous Advantage Actor Critic (A3C) algorithm. The algorithm can be implemented to predict both value function and optimal policy functions as is done here using Actor Critic.

ACKNOWLEDGMENT

This research paper would not have been possible without the exceptional support of my mentor, Dr. Anitha V who is a wonderful inspiration. Her enthusiasm, knowledge share, domain expertise, impeccable inputs and willingness to give her time have been valuable. The authors would like to thank her for all her contributions.

BIOGRAPHIES

Shruthi N is a Ph.D. research scholar in Bangalore, Karnataka, India under Visvesvaraya Technological University, 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 6 International Journal publications to her credit.

Dr.Siddesh.G.K. is Head, Research and Development at New Horizon College of Engineering, Bangalore. He received a Bachelor's degree in Electronics Communication Engineering from Bangalore University in 1998, M.Tech. in Digital Electronics and Advanced Communications from Manipal Institute of Technology, Manipal, Karnataka in 2002 and Ph.D.in Electronics Communication Engineering from Visvesvaraya Technological University, Belagavi in 2013.

His work experience includes academic, research administration of more than 20+ years in various engineering colleges. He has published more than 45 research papers in various National, International Journals and Conferences in India and abroad. He also has book chapters from reputed publishers to his credit.

DECLARATION

Compliance with Ethical Standards: The authors declare that the submitted manuscript has not been published elsewhere. No funding was received to assist with the preparation of this manuscript. All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript. Competing Interests: The authors have no competing interests to declare that are relevant to the content of this article.

Research Data Policy Data Availability: The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors have no competing interests to declare that are relevant to the content of this article. There is no conflict of interest.

REFERENCES

[1] J. Chen and X. Ran, "Deep learning with edge computing: A review.," Proc. IEEE, vol. 107, no. 8, pp. 1655–1674, 2019.

[2] 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.

[3] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: Convergence of computing, caching and communications," Ieee Access, vol. 5, pp. 6757–6779, 2017.

[4] Fangxin Wang, Miao Zhang, X.Wang, X.Ma, J.Liu, "Deep Learning for Edge Computing Applications: A State-of-the-Art Survey, "IEEE access, vol. 8, pp. 58322-58336, March 23 2020.

[5] 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.

[6] L. Xiao, X. Wan, C. Dai, X. Du, X. Chen, and M. Guizani, "Security in mobile edge caching with reinforcement learning," IEEE Wireless Communications, vol. 25, no. 3, pp. 116–122, 2018.

[7] B. Cao, L. Zhang, Y. Li, D. Feng, and W. Cao, "Intelligent offloading in multi-access edge computing: A state-of-the-art review and framework," IEEE Communications Magazine, vol. 57, no. 3, pp. 56–62, 2019.

[8] A. Marchisio, M. A. Hanif, F. Khalid, G. Plastiras, C. Kyrkou,T.Theocharides, and M. Shafique,"Deep learning for edge computing:Current trends, cross-layer optimizations, and open research challenges," Research Gate, IEEE, 2019.

[9] M. Zhang, F. Zhang, N. D. Lane, Y. Shu, X. Zeng, B. Fang, S. Yan, and H. Xu, "Deep learning in the era of edge computing: Challenges and opportunities," Fog Computing: Theory and Practice, pp. 67–78, 2020.

[10] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. MIT press, 2016.

[11] Z. Khan, M. Chowdhury, M. Islam, C.-Y. Huang, and M. Rahman, "Long short-term memory neural networks for false information attack detection in software-defined in-vehicle network," arXiv preprint arXiv:1906.10203, 2019.

[12] T. T. Nguyen and V. J. Reddi, "Deep reinforcement learning for cyber security," arXiv preprint arXiv:1906.05799, 2019.

[13] 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.

[14] G. Joseph, M. C. Gursoy, and P. K. Varshney, "Anomaly detection under controlled sensing using actor-critic reinforcement learning," in 2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), pp. 1–5, IEEE, 2020.

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

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

[17] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, Applied logistic regression, vol. 398. John Wiley & Sons, 2013.

[18] D. Kartik, E. Sabir, U. Mitra, and P. Natarajan, "Policy design for active sequential hypothesis testing using deep learning," in 2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 741–748, IEEE, 2018.

[19] X. Huang, L. He, X. Chen, L. Wang, and F. Li, "Revenue and energy efficiency-driven delay constrained computing task offloading and resource allocation in a vehicular edge computing network: A deep reinforcement learning approach," arXiv preprint arXiv:2010.08119, 2020.

[20] G. Joseph, M. C. Gursoy, and P. K. Varshney, "A scalable algorithm for anomaly detection via learning-based controlled sensing," arXiv preprint arXiv:2105.06289, 2021.