Neurocomputing 396 (2020) 406–428

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Deep Learning Clusters in the Cognitive Packet Network

Will Serrano*, Erol Gelenbe

Intelligent Systems and Networks Group, Electrical and Electronic Engineering, Imperial College London, United Kingdom

ARTICLE INFO

Article history:
Received 6 February 2018
Revised 25 May 2018
Accepted 25 July 2018
Available online 24 April 2019

Keywords:
Random Neural Network
Deep Learning Clusters
Cognitive Packet Network
QoS
Cybersecurity
Routing

ABSTRACT

The Cognitive Packet Network (CPN) bases its routing decisions and flow control on the Random Neural Network (RNN) Reinforcement Learning algorithm; this paper proposes the addition of a Deep Learning (DL) Cluster management structure to the CPN for Quality of Service metrics (Delay Loss and Bandwidth), Cyber Security keys (User, Packet and Node) and Management decisions (QoS, Cyber and CEO). The RNN already models how neurons transmit information using positive and negative impulsive signals whereas the proposed additional Deep Learning structure emulates the way the brain learns and takes decisions; this paper presents a brain model as the combination of both learning algorithms, RNN and DL. The proposed model has been simulated under different network sizes and scenarios and it has been validated against the CPN itself without DL clusters. The simulation results are promising; the presented CPN with DL clusters as a mechanism to transmit, learn and make packet routing decisions is a step closer to emulate the way the brain transmits information, learns the environment and takes decisions.

1. Introduction

Our brain performs several functions at the same time; it learns about the environment from our five senses; it stores memories to preserve our identity; it takes decisions on different situations and finally; it protects itself against external treats or attacks. Our brain is formed by clusters of neurons [1] specialized in learning from different senses where information is transmitted as positive and negative spikes or impulses. It functions with two types of memories [2]; short term memory is used for fast decisions and task related actions whereas long term memory preserves our identity and security. Another brain duality consists on its two operation modes [3]; consciousness under normal activities and unconsciousness under emergency situations such as being under external attack or routine operations like storing information while sleeping.

1.1. Related work

The expansion of the connectivity provided by the Ethernet and Internet protocols has enabled new industrial, technological and social applications and services however users are increasingly under new cybersecurity threats and risks. Ericsson [4] introduces cybersecurity issues and threats within Power Communications Systems in a smart grid infrastructure where network vulnerabilities and information security domains are analyzed. Ten et al. [5] present a survey on cybersecurity of critical infrastructure; in addition they propose a SCADA framework based on four procedures: real time monitoring, anomaly detection, impact analysis and mitigation strategy. They model an attack tree analysis with an algorithm for cybersecurity evaluation that incorporates password policies and port auditing. Cruz et al. [6] present a distributed intrusion detection system for SCADA systems that includes different types of security agents tuned for each specific domain: development of network, device and process level capabilities, integration of signature and anomaly based techniques against threats and finally the adoption of a distributed multi layered design with message queues to transmit predefined events between elements. Wang et al. [7] propose a framework to facilitate the development of adversary resistant Deep Neural Networks (DNN) by inserting a data transformation module between the sample and the DNN that avoids threat samples with a minimum impact on the classification accuracy. Tuor et al. [8] present an unsupervised Deep Learning approach to detect anomalous network activity from system logs in real time where events are extracted as features and the DNN learns users' normal behaviour or anomaly as potential malicious behaviour. Wu et al. [9] present a classification of cyber physical attacks and risks in cyber manufacturing systems with possible mitigation measures such as supervised machine learning for classification and unsupervised machine learning for anomaly detection on physical data. Kim [10] proposes a new cyber defensive computer control system architecture based on the diversification of hardware systems and unidirectional communications assuming that the detection and prevention of cyber attacks will never be complete.

* Corresponding author.
E-mail addresses: g.serrano11@imperial.ac.uk (W. Serrano), e.gelenbe@imperial.ac.uk (E. Gelenbe).

https://doi.org/10.1016/j.neucom.2018.07.101
0925-2312/

Deep Learning is characterized for using a cascade of L-layers of non linear processing units for feature extraction and transformation; each successive layer uses the output from the previous layer as input. Deep Learning learns multiple layers of representations that correspond to different levels of abstractions; those levels form a hierarchy of concepts where the higher the level, the more abstract concepts are learned. Schmidhuber [24] examines Deep Learning in neural networks. Bengio et al. [25] review recent work in the area of unsupervised feature learning and Deep Learning including advances in probabilistic models. They propose a new probabilistic framework to include likelihood based probabilistic models, reconstruction based models such as auto encoder variants and geometrically based manifold learning approaches. Jiea et al. [26] propose a progressive framework to deep optimize neural networks. They combine the stability of linear methods with the ability of learning complex and abstract internal representations of Deep Learning methods. They insert a linear loss layer between the input layer and the first hidden non-linear layer of a traditional Deep Learning model. Le et al. [27] study the advantages and disadvantages of off-the shelf optimization algorithms in the context of simplification and speed up the process of pre-training the unsupervised feature learning and Deep Learning. Ngiam et al. [28] propose an application of deep networks to learn features over multiple modalities to demonstrate that cross modality feature learning performs better than single modality learning. Sutskever et al. [29] present an approach to sequence learning that makes minimal assumptions on the sequence structure using a multilayered Long Short Term Memory (LSTM) to map the input sequence to a vector of a fixed dimensionality. Bekker et al. [30] propose an intra cluster training strategy for Deep Learning with applications to language identification where the language clusters are used to define a cost function to train a neural network.

The concept of Cognitive Packet Networks or Artificial Intelligence in network routing has also been researched. Li and Zhang [37] define the architecture of Network Artificial Intelligence (NAI) that includes key components and key protocol extension requirements for self-adjustment, self-optimization, self-recovery of the network through collection of Big data of network state and machine learning. Zhang et al. [38] propose a collaborative Internet architecture that removes the restrictions from the resource/location binding, user/network binding, and control/data binding, which are the root causes of the current Internet's issues. Qadir et al. [39] provide a vision how Artificial Intelligence can simplify network management such as cloud computing, network functions virtualization, and software-defined networking where intelligent services and cognitive networks will show network-wide intelligent behaviour to solve problems of network heterogeneity, performance, and quality of service (QoS). Quan et al. [40] investigate a new Smart Identifier NETworking (SINET) prototype and propose a customized solution that enables crowd collaborations for software defined vehicular networks through crowd sensing where network function allocations are organized with a group of components with similar function.

1.2. Summary of contributions

This paper presents the association of the most complex biological system; our brain with the most complex artificial system represented in large data networks: the Internet; the information infrastructure of the Big Data and the Web. The link between both of them is the Random Neural Network [16–18]. Data networks collect information from users and transmit it to different locations; to perform this activity, they are required to make routing decisions based on different Quality of Service metrics while storing routing tables in memory under the threat of Cyber attacks.

This paper proposes the Cognitive Packet Network (CPN) [11–15] with an additional Deep Learning (DL) cluster [31,32] structure that emulates how the brain operates. The proposed model adds a layer of specialised Deep Learning management clusters that take the final routing decision; DL clusters behave as a long term memory to remember network identity: QoS metrics and Cyber keys. The CPN-RNN routing algorithm is chosen under normal or conscious operations due its fast and adaptable route learning as short memory whereas DL cluster route is selected when the network is under external cyber attacks. DL clusters take routing decisions based on the long term memory in unconsciousness operation as a safe and resilient although inefficient and inflexible routing.

The mathematical model of CPN with DL clusters is described in Section 2. The implementation of the CPN-DL is defined in Section 3. The validation of the proposed model under different QoS and Cyber scenarios in small (3×3, 4×4, 5×5), medium (6×6, 7×7) and large square configuration node networks (8×8, 9×9, 10×10) from one up to eight decision layers, respectively, is presented in Section 4. Final conclusions are presented in Section 5, and related bibliography is presented at the endof the references.

2. The Cognitive Packet Network with Deep Learning Clusters

The Cognitive Packet Network was introduced by Gelenbe et al. [11–15]; it has been tested in large scale networks up to 100 nodes with worst and best case performance scenarios. The CPN assigns routing and flow control capabilities to the packets rather than the nodes. QoS goals are assigned to Cognitive Packets (CP) within the CPN, which they follow when making routing decisions themselves with minimum dependence on the nodes.

Given a Goal G based on QoS parameters that the CP has to achieve, $G = \alpha\text{Delay} + \beta\text{Loss} + \gamma\text{Bandwidth}$, and its associated reward R which is $R = 1/G$. Successive measured values of the R are denoted by R_l, $l = 1,2\ldots$ These are used to compute a decision threshold $T_l = \alpha T_{l\text{-}1} + (1\text{-}\alpha)R_l$. The CP makes a routing decision based on this value; if the observed measured reward is greater than the associated node threshold; the CPN rewards the decision taken; otherwise; it penalises it (Fig. 1).

The Random Neural Network [16–18] represents more closely how signals are transmitted in many biological neural networks where they travel as spikes or impulses, rather than as analogue signal levels. The RNN is a spiking recurrent stochastic model for neural networks where its main analytical properties are the "product form" and the existence of the unique network steady state solution. It has been applied in different applications including network routing in the Cognitive Packet Network with Reinforcement Learning algorithm, which requires the search for paths that meet certain pre-specified Quality of Service requirements

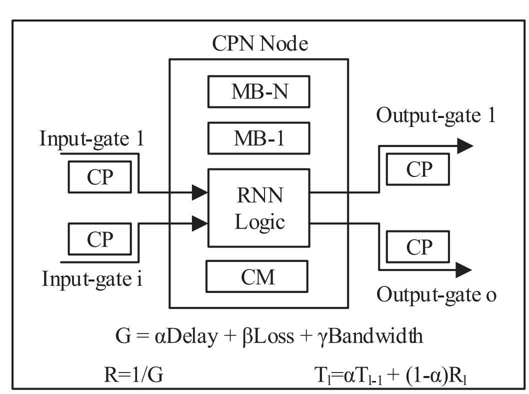


Fig. 1. The Cognitive Packet Network.

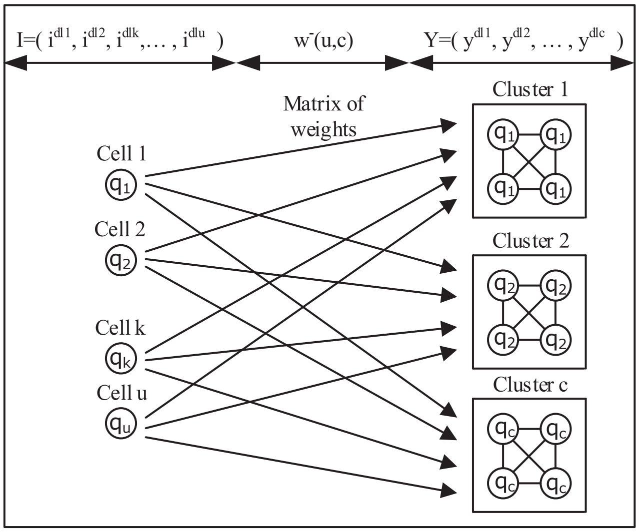


Fig. 2. The Random Neural Network with multiple clusters.

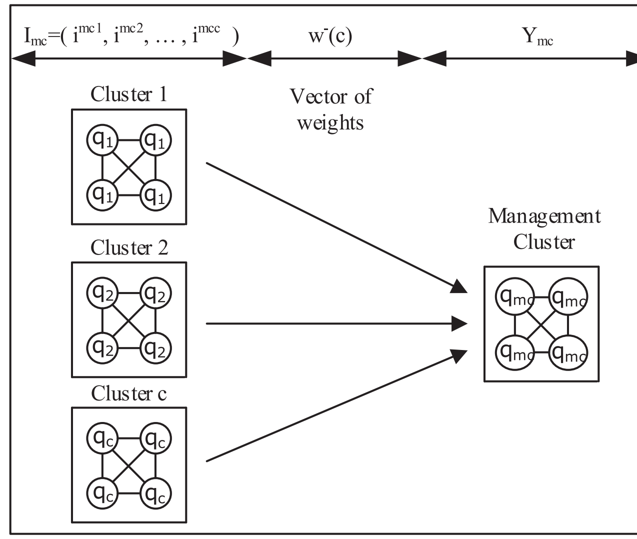


Fig. 3. The Random Neural Network with a Management Cluster.

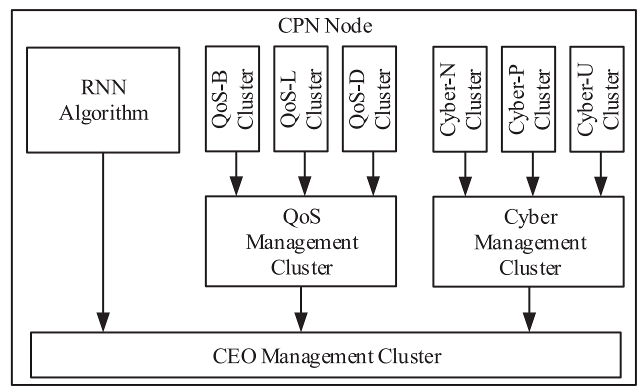


Fig. 4. CPN node with DL clusters architecture.

[11–15], search for exit routes for evacuees in emergency situations [19,20], pattern based search for specific objects [21], video compression [22], and image texture learning and generation [23].

Deep Learning with the Random Neural Network is described by Gelenbe and Yin [31,32]. This model is based on the generalized queuing network with triggered customer movement (G-networks) where customers are either "positive" or "negative" that can be moved from queues or leave the network. G-Networks are introduced by Gelenbe, E. [33,34]; an extension to this model is developed by Gelenbe, E. et al. [35] where synchronized interactions of two queues could add a customer in a third queue (Fig. 2).

The Deep Learning Clusters model defines:

- a U-dimensional vector I ∈ $[0,1]^U$ that represents the input state $\overline{q_u}$ for the cell u:

$$\mathrm{I} = \left(\mathrm{i}^{\mathrm{dl}_1},\ \mathrm{i}^{\mathrm{dl}_2},\ \mathrm{i}^{\mathrm{dl}_k},\ \ldots,\ \mathrm{i}^{\mathrm{dl}_u}\right) \tag{1}$$

- the $U \times C$ matrix of weights from the U input cells to the cells in each of the C clusters:

$$\mathrm{w}^{-}(\mathrm{u},\ \mathrm{c}) \tag{2}$$

- a C-dimensional vector $Y \in [0,1]^C$ that represents the cell state q_c for the cluster c:

$$\mathrm{Y} = \left(\mathrm{y}^{\mathrm{dl}_1},\ \mathrm{y}^{\mathrm{dl}_2},\ \ldots,\ \mathrm{y}^{\mathrm{dl}_c}\right) \tag{3}$$

The network learns the $U \times C$ weight matrix $w^{-}(u,c)$ by calculating the new values of the network parameters for the input I and output Y using Gradient Descent learning algorithm which optimizes the network weight parameters $w^{-}(u,c)$ from a set of input-output pairs $(i^{dl}{}_u, y^{dl}{}_c)$.

The Deep Learning management cluster was proposed by Serrano and Gelenbe [36]. It takes management decisions based on the inputs from different Deep Learning clusters (Fig. 3).

The Deep Learning management cluster model considers:

- a C-dimensional vector $I_{mc} \in [0,1]^C$ that represents the input state $\overline{q_c}$ for cluster c:

$$\mathrm{I}_{\mathrm{mc}} = \left(\mathrm{i}^{\mathrm{mc}_1},\ \mathrm{i}^{\mathrm{mc}_2},\ \ldots,\ \mathrm{i}^{\mathrm{mc}_c}\right) \tag{4}$$

- the C-dimensional vector of weights from the C input clusters to the cells in the Management Cluster mc:

$$\mathrm{w}^{-}(\mathrm{c}) \tag{5}$$

- a scalar $Y_{mc} \in [0,1]$, the cell state q_{mc} for the Management Cluster mc representing its final decision:

$$\mathrm{Y}_{\mathrm{mc}} \tag{6}$$

2.1. The Cognitive Packet Network with Deep Learning Clusters model

The CPN instantaneously updates its network weights based on the direct observations from the network parameters; this enables its routing algorithm to take fast decisions adaptable to QoS changes. The CPN emulates the brain in conscious mode when taking fast decisions in normal operation using short term memory based on the direct information received from the senses. This paper proposes the addition of a Deep Learning Cluster structure to the Cognitive Packet Network where each DL cluster learns different QoS network metrics (Delay, Loss and Bandwidth), the best routes for each QoS metric, and Cyber keys (User, Packet and Node). In addition, this paper proposes the addition of a layer of DL management clusters (QoS, Cyber and CEO) that take the final routing decision based on the inputs from the DL QoS clusters and CPN-RNN algorithm. The Deep Learning routing algorithm adapts slowly to network changes where the proposed model applies it as a reliable and safe routing when the CPN is compromised by a Cyber attack; it emulates the brain in subconscious mode using long term memory when it takes minimum decisions for defense or survival (Fig. 4).

2.2. QoS Deep Learning Cluster

A Deep Learning Cluster is assigned to each QoS metric: Delay, Packet Loss and Bandwidth. Each QoS DL cluster learns the best associated QoS metric with its best associated node gates. When a node observes a better QoS route with a lower QoS metric; it learns its value and includes the gate on the first position of the QoS DL routing table.

This model defines three QoS clusters; Delay, Packet Loss and Bandwidth:

- a U-dimensional I_{QoS} ∈ $[0,1]^U$ vector where $i^{QoS}{}_1$, $i^{QoS}{}_2$, and $i^{QoS}{}_u$ are the same value for each QoS type:

$$I_{QoS} = \left(i^{QoS_1},\ i^{QoS_2}, \ldots, i^{QoS_u}\right) \tag{7}$$

- the $U \times C$ matrix of weights of the QoS Deep Learning Cluster:

$$w^-_{QoS}(u, c) \tag{8}$$

a C-dimensional vector Y_{QoS} ∈ $[0,1]^C$ where $y^{QoS}{}_1$ is the QoS metric and $y^{QoS}{}_2$, …, $y^{QoS}{}_c$ are the node's QoS best routing gates:

$$Y_{QoS} = \left(y^{QoS_1},\ y^{QoS_2}, \ldots, y^{QoS_c}\right) \tag{9}$$

2.3. Cyber Deep Learning Cluster

A Deep Learning Cluster is assigned per Cyber key: User, Packet and Node. The user cyber network weights authenticate the application that has transmitted the packet. The packet cyber network weights validates the packet transmitted is legitimate; this secures the network against Denial of Service attacks. The node cyber network weights authenticate the nodes within the CPN; this secures the CPN against impostor nodes. The Cyber network weights could have been assigned previously to the CPN nodes by the network administrator or the CPN nodes could have learnt them in an initialization mode.

When a CPN node receives a Cognitive Packet (CP); each Cyber DL cluster extracts its relevant keys and uses them as input and output values. If the quadratic error between the Cyber DL cluster output vector and the input vector is over a threshold then the CPN node considers the certificate as invalid or the CPN is under Cyber attack.

This model defines there Cyber clusters; User Packet and Node:

- a *U*-dimensional vector I_{Cyber} ∈ $[0,1]^U$ where $i^{Cyber}{}_1$, $i^{Cyber}{}_2$, …, $i^{Cyber}{}_u$ are the Cyber keys from the CP:

$$I_{Cyber} = \left(i^{Cyber_1},\ i^{Cyber_2}, \ldots, i^{Cyber_u}\right) \tag{10}$$

- the *U* x *C* matrix of weights of the Cyber Deep Learning Cluster:

$$w^-_{Cyber}(u,\ c) \tag{11}$$

- a C-dimensional vector Y_{Cyber} ∈ $[0,1]^C$ where $y^{Cyber}{}_1$, $y^{Cyber}{}_2$, …, $y^{Cyber}{}_c$ are the Cyber keys from the DL cluster:

$$Y_{Cyber} = \left(y^{Cyber_1},\ y^{Cyber_2}, \ldots, y^{Cyber_c}\right) \tag{12}$$

2.4. Deep Learning management cluster

The DL management clusters take the overall routing decision. The QoS and Cyber management clusters analyse the output from their associated QoS and Cyber DL clusters respectively (Fig. 5). If the Cyber management cluster detects a failure in the cyber certificates; the CEO management cluster routes the network Cognitive Packets as safe mode using the QoS DL clusters, otherwise, if the Cyber certificates are valid the CEO management cluster choses the route provided by the CPN-RNN routing algorithm as normal mode.

This model defines the QoS management cluster as:

- a C-dimensional vector I_{qmc} ∈ $[0,1]^C$ with the values of the QoS Metrics for each QoS cluster:

$$I_{qmc} = \left(i^{qmc_1},\ i^{qmc_2}, \ldots, i^{qmc_c}\right) \tag{13}$$

- the C-dimensional vector of weights that represents the Goal = (α_{Delay}, β_{Loss}, $\gamma_{Bandwidth}$):

$$w^-_{qmc}(c) \tag{14}$$

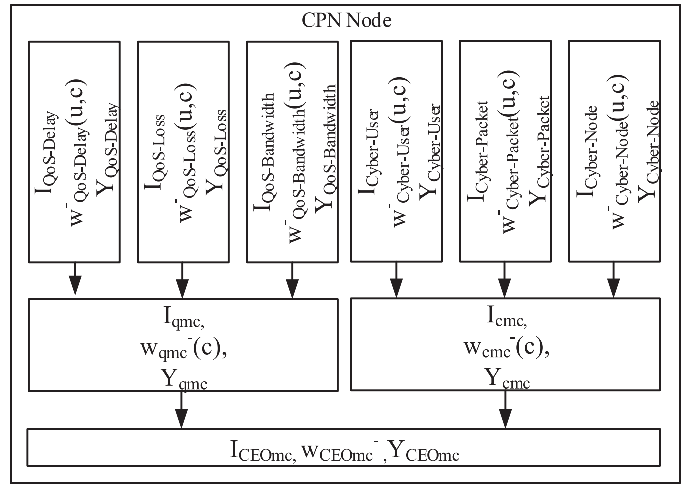


Fig. 5. CPN node with DL clusters model.

- a scalar Y_{qmc} ∈ [0,1] that represents the best QoS metric routing decision to be taken:

$$Y_{qmc} \tag{15}$$

Cyber management cluster as:

- a C-dimensional vector I_{cmc} ∈ $[0,1]^C$ with the values of the key errors for each Cyber cluster (User, Packet, Node):

$$I_{cmc} = \left(i^{cmc_1},\ i^{cmc_2}, \ldots, i^{cmc_c}\right) \tag{16}$$

- the C-dimensional vector of weights that represents the relevance of each Cyber Cluster:

$$w^-_{cmc}(c) \tag{17}$$

- a scalar Y_{cmc} ∈ [0,1] that represents if the packet has passed the Cyber network security:

$$Y_{cmc} \tag{18}$$

CEO management cluster as:

- a scalar I_{CEOmc} ∈ [0,1] with the values of the QoS management cluster:

$$I_{CEOmc} \tag{19}$$

- a scalar $w_{CEOmc}{}^-$ ∈ [0,1] that represents the error of the Cyber management cluster:

$$w^-_{CEOmc} \tag{20}$$

- a scalar Y_{CEOmc} ∈ [0,1] that represents the final routing decision:

$$Y_{CEOmc} \tag{21}$$

3. Implementation

The Cognitive Packet Network with Deep Learning Clusters is implemented in the Network Simulator Omnet 5.0. The simulation covers several size $n \times n$ square networks where all the nodes in the same and adjacent layers are connected with each other. For simplicity, the simulation always consider the first node (Node 1) as the only transmitter and the last node (Node n) as the only receiver; the other nodes only participate in the routing of Cognitive Packets. An example of a 4×4 network is shown in Fig. 6.

Each node has normalized QoS Delay, Loss and Bandwidth metrics as relative to their number; in a $n \times n$ network node *i* will

Table 1
QoS values – 4 × 4 network.

Node 4 Initial - Final	**Node 5 Initial - Final**	**Node 9 Initial - Final**	**Node 16 Initial - Final**
Delay: 40 – 40	Delay: 50 – 80	Delay: 90–120	Delay: 160–160
Loss: 65 – 65	Loss: 60 – 45	Loss: 40 –25	Loss: 05–05
Bandwidth: 45 – 45	Bandwidth: 55 – 85	Bandwidth: 95 – 125	Bandwidth: 165 – 165
Node 3 Initial - Final	**Node 6 Initial - Final**	**Node 10 Initial - Final**	**Node 15 Initial - Final**
Delay: 30 – 30	Delay: 60 – 70	Delay: 100 – 110	Delay: 150 – 150
Loss: 70 – 70	Loss: 55 – 50	Loss: 35 – 30	Loss: 10 – 10
Bandwidth: 35 – 35	Bandwidth: 65 – 75	Bandwidth: 105 – 115	Bandwidth:155 - 155
Node 2 Initial - Final	**Node 7 Initial - Final**	**Node 11 Initial - Final**	**Node 14 Initial - Final**
Delay: 20 – 20	Delay: 70 – 60	Delay: 110 – 100	Delay: 140 – 140
Loss: 75 – 75	Loss: 50 – 55	Loss: 30 – 35	Loss: 15 – 15
Bandwidth: 25 – 25	Bandwidth: 75 – 65	Bandwidth: 115 – 105	Bandwidth: 145 - 145
Node 1 Initial - Final	**Node 8 Initial - Final**	**Node 12 Initial - Final**	**Node 13 Initial - Final**
Delay: 10 – 10	Delay: 80 – 50	Delay: 120 – 90	Delay: 130 – 130
Loss: 80 – 80	Loss: 45 – 60	Loss: 25 – 40	Loss: 20 – 20
Bandwidth: 15 - 15	Bandwidth: 85 - 55	Bandwidth: 125 - 95	Bandwidth: 135 - 135

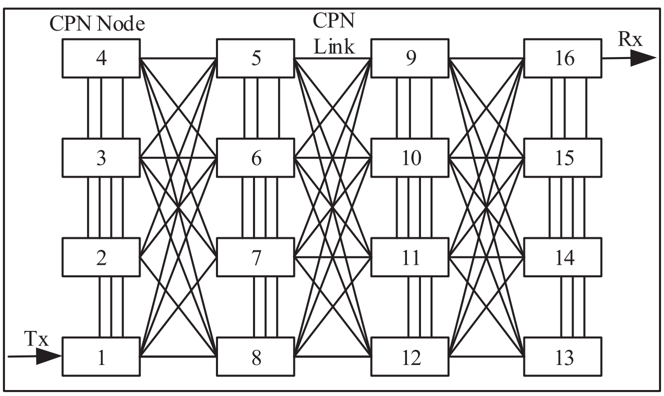


Fig. 6. 4 × 4 Node CPN-DL Network.

have Delay: i*10; Loss: $(n\text{-}i)$*5 and Bandwidth: 5+(i*10), respectively. The approach is represented in the Table 1 for a 4 × 4 network. After two Cognitive Packets are sent with a defined QoS; the QoS metric swaps between each internal node the within the same column as shown on the Table 1 for a 4 × 4 network. This model proposes to set the CPN-RNN network weights with initialization packets sent at random gates.

3.1. QoS DL clusters

The QoS DL clusters have three input cells ($u = 3$) and three output clusters ($c = 3$). The model therefore has $i^{\text{QoS-d}}{}_1$=0.5; $i^{\text{QoS-d}}{}_2$=0.5 and $i^{\text{QoS-d}}{}_3$=0.5; $y^{\text{QoS-d}}{}_1$ is the best QoS Delay metric, $y^{\text{QoS-d}}{}_2$ the best QoS Delay route and $y^{\text{QoS-d}}{}_3$ the second best Delay route (Table 2). The model follows a similar approach for the Loss and Bandwidth QoS DL clusters respectively. The model normalizes the inputs of the DL clusters to (0.5+ QoS Metric/1000) and (0.5+ Best Gate/100), respectively.

Table 2
QoS Deep Learning Cluster Implementation.

Cluster	Input	Value	Output	Value
QoS Delay	$i^{\text{QoS-d}}{}_1$	0.5	$y^{\text{QoS-d}}{}_1$	Best QoS Delay Metric
QoS Delay	$i^{\text{QoS-d}}{}_2$	0.5	$y^{\text{QoS-d}}{}_2$	Best QoS Delay Gate
QoS Delay	$i^{\text{QoS-d}}{}_3$	0.5	$y^{\text{QoS-d}}{}_3$	Second Best QoS Delay Gate
QoS Loss	$i^{\text{QoS-l}}{}_1$	0.6	$y^{\text{QoS-l}}{}_1$	Best QoS Loss Metric
QoS Loss	$i^{\text{QoS-l}}{}_2$	0.6	$y^{\text{QoS-l}}{}_2$	Best QoS Loss Gate
QoS Loss	$i^{\text{QoS-l}}{}_3$	0.6	$y^{\text{QoS-l}}{}_3$	Second Best QoS Loss Gate
QoS Bandwidth	$i^{\text{QoS-b}}{}_1$	0.7	$y^{\text{QoS-b}}{}_1$	Best QoS Bandwidth Metric
QoS Bandwidth	$i^{\text{QoS-b}}{}_2$	0.7	$y^{\text{QoS-b}}{}_2$	Best QoS Bandwidth Gate
QoS Bandwidth	$i^{\text{QoS-b}}{}_3$	0.7	$y^{\text{QoS-b}}{}_3$	Second Best QoS Bandwidth Gate

Table 3
Cyber Deep Learning Cluster Implementation.

Cluster type	Input-Output	Value (Input =Output)
User	$i^{\text{Cyber-u}}{}_1 \dots i^{\text{Cyber-u}}{}_{10}$ $y^{\text{Cyber-u}}{}_1 \dots y^{\text{Cyber-u}}{}_{10}$	(0.9, 0.8, 0.9, 0.8, 0.9, 0.8, 0.9, 0.8, 0.9, 0.8)
Packet	$i^{\text{Cyber-p}}{}_1 \dots i^{\text{Cyber-p}}{}_{10}$ $y^{\text{Cyber-p}}{}_1 \dots y^{\text{Cyber-p}}{}_{10}$	(0.7, 0.6, 0.7, 0.6, 0.7, 0.6, 0.7, 0.6, 0.7, 0.6)
Node	$i^{\text{Cyber-n}}{}_1 \dots i^{\text{Cyber-n}}{}_{10}$ $y^{\text{Cyber-n}}{}_1 \dots y^{\text{Cyber-n}}{}_{10}$	(0.5, 0.4, 0.5, 0.4, 0.5, 0.4, 0.5, 0.4, 0.5, 0.4)

3.2. Cyber DL clusters

The Cyber DL clusters have ten input cells ($u = 10$) and ten output clusters ($c = 10$). The key is a vector of 10 dimensions. $i^{\text{Cyber-u}}{}_u$, $i^{\text{Cyber-p}}{}_{\text{u}}$, $i^{\text{Cyber-n}}{}_u$ have a value between 0.1 and 0.9 with increments 0.1Δ (Table 3). The Cyber DL clusters network weights are trained with the value of the input the same as the output.

3.3. DL management clusters

The inputs of the Cyber management cluster are the errors provided by each Cyber DL cluster and the value of its network weights are set with same value (0.1) therefore the different cyber DL clusters have the same priority. The output Y_{cmc} is the overall Cyber quantified error decision based on a threshold.

The inputs of the QoS management cluster are the best QoS metrics from each QoS DL cluster and the value of its networks weights corresponds to the Goal = (α_{Delay}, β_{Loss}, $\gamma_{\text{Bandwidth}}$). The output Y_{qmc} is quantified best QoS metric decision.

The input of the CEO management cluster is the value provided by the QoS management cluster and its network weight is the value provided by the Cyber management cluster. The output is the final routing decision between the different gates provided by the RNN algorithm, Delay, Loss and Bandwidth DL clusters respectively (Table 4).

The values of the different parameters for the Cyber and QoS Deep Learning management clusters (I, w and Y) are obtained respectively from the Cyber and QoS Deep Learning Clusters. Successively, the parameters of the CEO Deep Learning management cluster are obtained from Cyber, QoS Deep Learning management clusters. The thresholds or ranges correspond to the DL Management cluster activation function [36]; this activation function is shown on Fig. 7 with explicit values for the CEO Deep Learning management cluster.

Table 4
Deep Learning Management Cluster Implementation.

Cluster	Input	Network weights	Output
Cyber	$\mathbf{I_{Cmc}}$ (User Error, Packet Error, Node Error)	$\mathbf{w_{Cmc}^{-}(c)}$ (0.1, 0.1, 0.1)	$\mathbf{Y_{Cmc}}$ 0.0 if $Y_{cmc} > 0.999$ (Normal Condition) 0.999 if $Y_{cmc} \leq 0.999$ (Cyber Attack)
QoS	$\mathbf{I_{Qmc}}$ (Delay Metric, Loss Metric, Bandwidth Metric)	$\mathbf{w_{Qmc}^{-}(c)}$ (α_{Delay}, β_{Loss}, $\gamma_{Bandwidth}$)	$\mathbf{Y_{Qmc}}$ 0.1 if $Y_{Delay\text{-}qmc} > Y_{Loss\text{-}qmc}$ and $Y_{Delay\text{-}qmc} > Y_{Bandwidth\text{-}qmc}$ 0.5 if $Y_{Loss\text{-}qmc} > Y_{Delay\text{-}qmc}$ and $Y_{Loss\text{-}qmc} > Y_{Bandwidth\text{-}qmc}$ 0.9 if $Y_{Bandwidth\text{-}qmc} > Y_{Delay\text{-}qmc}$ and $Y_{Bandwidth\text{-}qmc} > Y_{Loss\text{-}qmc}$
CEO	$\mathbf{I_{CEOmc}}$ (0.1, 0.5 or 0.9)	$\mathbf{w_{CEOmc}^{-}}$ (0.0 or 0.999)	$\mathbf{Y_{CEOmc}}$ CPN-RNN if $0.6 < Y_{CEOmc} < 1$ DL-Delay if $0.4 < Y_{CEOmc} < 0.6$ DL-Loss if $0.2 < Y_{CEOmc} < 0.4$ DL-Bandwidth if $0.1 < Y_{CEOmc} < 0.2$

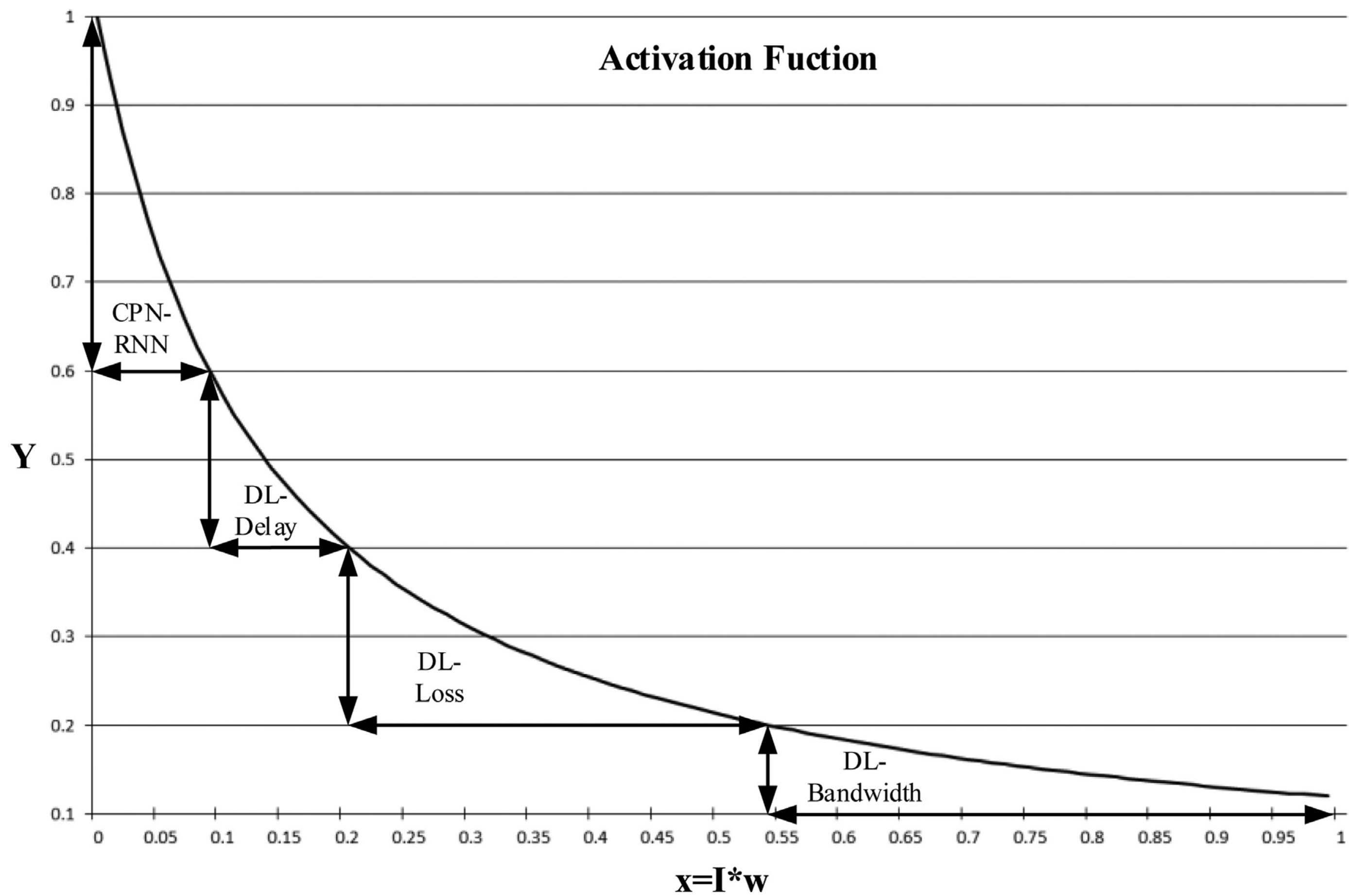


Fig. 7. Deep Learning Management Cluster activation function.

4. Experimental results

Different square $n \times n$ node network sizes are simulated, from 3×3 up to 10×10 with different Cyber keys; QoS metrics and Goal changes to assess the adaptability and performance of our proposed solution.

4.1. Cyber DL cluster validation

The different Cyber DL clusters are validated where the security keys are modified at node 1 and the cyber validation error is measured at the next node 4 once the CPs have an stable route. The keys are gradually changed; from the correct key to 0.1Δ increments applied to the different key dimensions. Table 5, Table 6 and Table 7 shows the values for the Cyber User, Packet and Node Deep Learning cluster validation respectively.

The Cyber DL cluster error largely increases even only with one 0.1Δ increment. The results are consistent between the different Cyber DL clusters. Cyber key increments have a bigger error if they are applied in the same dimension rather than split into different dimensions.

Table 5
Cyber User Deep Learning Cluster Validation.

Dimension	Δ=0.0	Δ=0.1	Δ=0.2	Δ=0.3	Δ=0.4
1	9.75E−11	0.0102	0.0409	0.0921	0.1638
2	9.7537E−11	0.0213	0.0851	0.1915	0.3406
3	9.7537E−11	0.0326	0.1305	0.2938	0.5226
4	9.7537E−11	0.0451	0.1806	0.4067	0.7238
5	9.7537E−11	0.0576	0.2306	0.5195	0.9249
6	9.7537E−11	0.0715	0.2867	0.6465	1.1519
7	9.7537E−11	0.0851	0.3414	0.7703	1.3732
8	9.7537E−11	0.1006	0.4038	0.9119	1.6273
9	9.7537E−11	0.1153	0.4633	1.0470	1.8698
10	9.7537E−11	0.1323	0.5321	1.2038	2.1526

Table 6 shows the values for the Cyber Packet Deep Learning Cluster validation.

Table 6
Cyber Packet Deep Learning Cluster Validation.

Dimension	Δ=0.0	Δ=0.1	Δ=0.2	Δ=0.3	Δ=0.4
1	4.72E−10	0.0108	0.0431	0.0970	0.1725
2	4.7238E−10	0.0233	0.0933	0.2104	0.3747
3	4.7238E−10	0.0373	0.1497	0.3382	0.6036
4	4.7238E−10	0.0533	0.2147	0.4864	0.8709
5	4.7238E−10	0.0707	0.2855	0.6488	1.1659
6	4.7238E−10	0.0904	0.3664	0.8363	1.5101
7	4.7238E−10	0.1112	0.4527	1.0379	1.8831
8	4.7238E−10	0.1347	0.5509	1.2701	2.3192
9	4.7238E−10	0.1592	0.6541	1.5160	2.7853
10	4.7238E−10	0.1866	0.7711	1.7993	3.3325

Table 7 shows the values for the Cyber Node Deep Learning Cluster validation.

Table 7
Cyber Node Deep Learning Cluster Validation.

Dimension	Δ=0.0	Δ=0.1	Δ=0.2	Δ=0.3	Δ=0.4
1	9.19E−10	0.0114	0.0458	0.1032	0.1838
2	9.1918E−10	0.0200	0.0800	0.1800	0.3200
3	9.1918E−10	0.0258	0.1259	0.2835	0.5044
4	9.1918E−10	0.0400	0.1600	0.3600	0.6400
5	9.1918E−10	0.0515	0.2061	0.4639	0.8250
6	9.1918E−10	0.0600	0.2400	0.5400	0.9600
7	9.1918E−10	0.0715	0.2863	0.6442	1.1455
8	9.1918E−10	0.0800	0.3200	0.7200	1.2800
9	9.1918E−10	0.0916	0.3664	0.8246	1.4661
10	9.1918E−10	0.1000	0.4000	0.9000	1.6000

4.2. 3 × 3 network – QoS DL cluster validation

The 3 × 3 network is simulated with a continuous 240 Cognitive Packet stream. The first 100 packets are used to initialize the CPN network. Goal changes after 20 packets whereas QoS metric changes 2 packets after the new Goal is selected following $T_l = 0.9^{*}T_{l\text{-}1} + 0.1^{*}R$ where T_l is the Threshold at decision packet l and R is the Reward at Node 1 (Table 8).

The QoS DL clusters have been validated with seven different variable Goals for the same Cognitive Packet stream. The CPN-RNN route decision taken by the CEO Management Cluster when the Cyber management cluster has authorised the different Cyber keys is shown on Fig. 8. The route provided by the QoS DL clusters remains unchanged due its slow learning process until the new best route is found by the CPN-RNN. The average Error and Iteration values for the different Deep Learning Clusters are represented on Table 9.

Table 10 shows the number of updates for the DL cluster and the CPN-RNN.

Table 8
3 × 3 Network QoS Deep Learning Cluster Validation – Simulation Parameters.

Cognitive packet	Goal	QoS metric
100	Network Initialization Cognitive Packets	
001–002	1.0*Delay+0.0*Loss+0.0* Bandwidth	Initial values
003–020	1.0*Delay+0.0*Loss+0.0* Bandwidth	Final values
021–022	0.0*Delay+1.0*Loss+0.0* Bandwidth	Initial values
023–040	0.0*Delay+1.0*Loss+0.0* Bandwidth	Final values
041–042	0.0*Delay+0.0*Loss+1.0* Bandwidth	Initial values
043–060	0.0*Delay+0.0*Loss+1.0* Bandwidth	Final values
061–062	0.5*Delay+0.5*Loss+0.0* Bandwidth	Initial values
063–080	0.5*Delay+0.5*Loss+0.0* Bandwidth	Final values
081–082	0.5*Delay+0.0*Loss+0.5* Bandwidth	Initial values
083–100	0.5*Delay+0.0*Loss+0.5* Bandwidth	Final values
101–102	0.0*Delay+0.5*Loss+0.5* Bandwidth	Initial values
103–120	0.0*Delay+0.5*Loss+0.5* Bandwidth	Final values
121–122	0 × 3Delay+0 × 3Loss+0.3* Bandwidth	Initial values
123–140	0 × 3Delay+0 × 3Loss+0.3* Bandwidth	Final values

The CPN-RNN algorithm continuously updates its network weighs whereas the DL Cluster route only refreshes when a better route is found, however the number of required iterations to update CPN-RNN is only one whereas DL clusters require approximately 165 iterations as shown on Table 9.

Fig. 9 shows the final CPN-DL route follows the Optimum Route in a 3 × 3 Network.

4.3. 3 × 3 node network – DL management cluster validation

The DL Management Clusters (Cyber, QoS and CEO) on this section are validated under two different Cyber Security scenarios; Δ=0: normal operation and Δ=0.1: CPN under Cyber attack. Three different strategic Cognitive Packets (CP 30, CP 85 and CP 148) are chosen for the 3 × 3 network validation with different Goals (Table 11).

4.4. 4 × 4 node network – QoS DL cluster validation

The 4 × 4 network is simulated with a continuous 380 Cognitive Packet stream. The first 100 packets are used to initialize the CPN network. Goal changes after 40 packets whereas QoS metric changes 2 packets after the new Goal is selected following T_l=0.99*$T_{l\text{-}1}$ + 0.01*R (Table 12).

The first two Cognitive Packets follow the best route whereas the third CP acknowledges the QoS metric has changed. The Threshold adapts progressively as the Goal degrades. Node 1 changes route after the CPN-RNN weights are updated finding the optimum route. When the new best route is discovered; the CPN Threshold adapts gradually to the original value. The average Error and Iteration values for the different Deep Learning Clusters are represented on Table 13.

Table 14 shows the number of updates for the DL cluster and the CPN-RNN.

The CPN-RNN algorithm continuously updates its network weighs whereas the DL Cluster route refreshes only when a better route is found, as the previous validation. The number of iterations to update CPN-RNN is only one whereas DL clusters require approximately 140 iterations as shown on Table 13 (Fig. 10).

The results provided by the 4 × 4 network are similar to the 3 × 3 network. The first two packets follow the best route whereas the third packet acknowledges the QoS metrics have changed. CPN-RNN finds the optimum route after Cognitive Packets explore the network and DL learns the route a Cognitive Packet after. Fig. 11 shows the CPN-DL route follows the Optimum Route in a 4 × 4 Network.

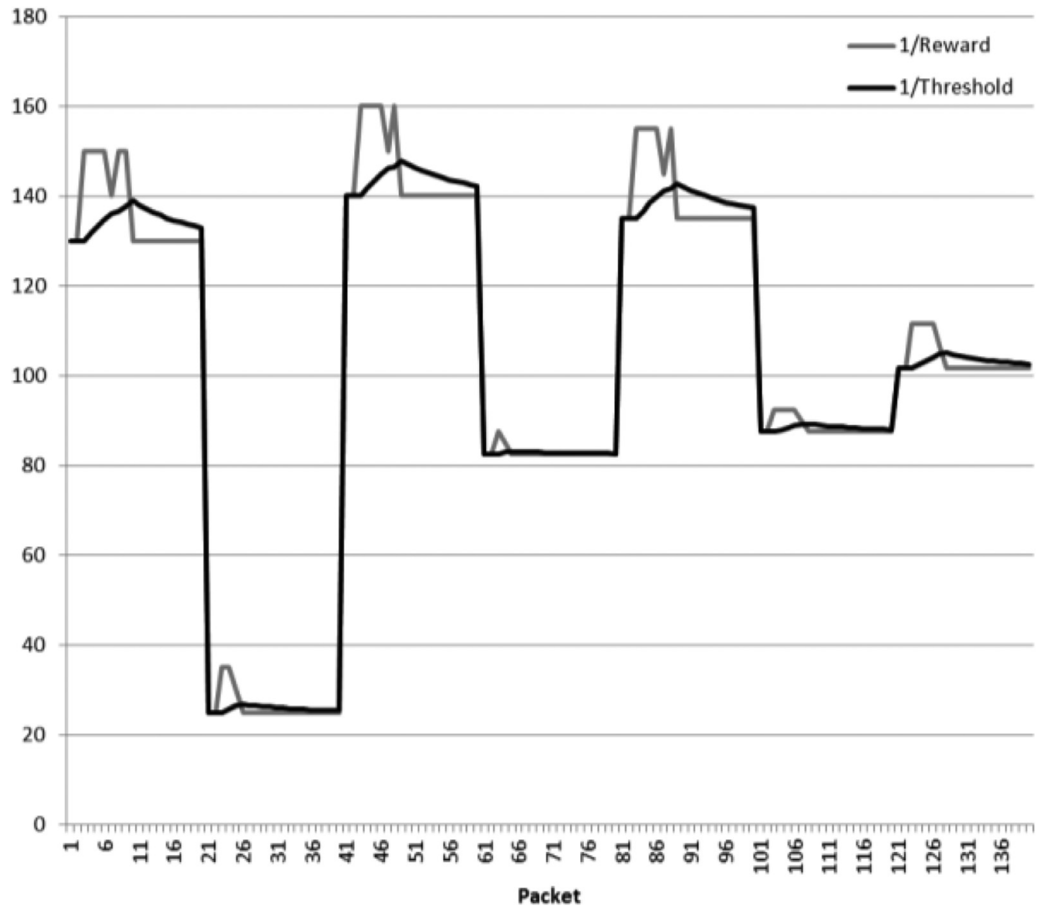


Fig. 8. 3×3 Network QoS Deep Learning Cluster validation.

Table 9
3×3 Network Deep Learning Cluster.

Average	Cyber user	Cyber packet	Cyber node	QoS delay	QoS loss	QoS bandwidth
Error	6.96E−10	7.34E−10	9.94E−10	9.47E−10	9.34E−10	9.54E−10
Iterations	58.00	108.00	1162.33	175.69	161.50	160.50

Table 10
3×3 Network Deep Learning Cluster vs CPN-RNN.

Updates	CPN-RNN	QoS delay	QoS loss	QoS bandwidth
Initialization	0	4	1	3
CP 001–140	140	9	1	9

4.5. 4 × 4 node network – DL management cluster validation

The results provided by the DL management cluster confirm the proposed model. The correct quantification of the DL management cluster cell states and the selection of the accurate thresholds are

Table 11
3×3 Network DL Management Cluster Validation.

Variable	Cognitive packet: 30 G:$1.0^*D+0.0^*L+0.0^*$B		Cognitive packet: 85 G: $0.5^*D+0.5^*L+0.0^*$B		Cognitive packet: 148 G: $0.3^*D+0.3^*L+0.3^*$B	
	Δ=0.0	Δ=0.1	Δ=0.0	Δ=0.1	Δ=0.0	Δ=0.1
Cyber I_{cmc}	5E−11	3.4E−4	5E−11	3.4E−4	5E−11	3.4E−4
Cyber Y_{cmc}	0.9994	0.9969	0.9994	0.9969	0.9994	0.9969
QoS-Delay I_{qmc}	0.6300	0.6300	0.3150	0.3150	0.2100	0.2100
QoS-Loss I_{qmc}	0.0000	0.0000	0.2625	0.2625	0.1750	0.1750
QoS-Band I_{qmc}	0.0000	0.0000	0.0000	0.0000	0.2133	0.2133
QoS-Delay Y_{qmc}	0.1765	0.1765	0.3000	0.3000	0.3913	0.3913
QoS-Loss Y_{qmc}	0.9994	0.9994	0.3396	0.3396	0.4354	0.4354
QoS- Band Y_{qmc}	0.9994	0.9994	0.9994	0.9994	0.3875	0.3875
CEO I_{CEOmc}	0.1000	0.1000	0.1000	0.1000	0.9000	0.9000
CEO $w_{CEOmc}^{-}(c)$	0.0000	0.9999	0.0000	0.9999	0.0000	0.9999
CEO Y_{CEOmc}	0.9994	0.5746	0.9994	0.5746	0.9994	0.1305
Routing decision	CPN Gate-4 Node 6	DL-Delay Gate-2 Node 4	CPN Gate-4 Node 6	DL-Delay Gate-2 Node 4	CPN Gate-4 Node 6	DL-Band Gate-2 Node 4

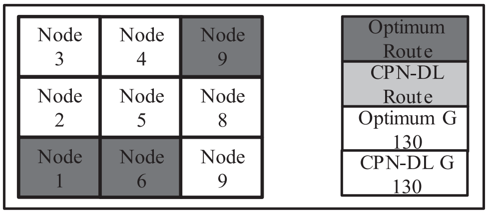


Fig. 9. 3 × 3 Network Final CPN-DL Route.

Table 12
4 × 4 Network QoS Deep Learning Cluster Validation – Simulation Parameters.

Cognitive packet	Goal	QoS metric
100	Network Initialization Cognitive Packets	
001–002	1.0*Delay+0.0*Loss+0.0* Bandwidth	Initial values
003–040	1.0*Delay+0.0*Loss+0.0* Bandwidth	Final values
041–042	0.0*Delay+1.0*Loss+0.0* Bandwidth	Initial values
043–080	0.0*Delay+1.0*Loss+0.0* Bandwidth	Final values
081–082	0.0*Delay+0.0*Loss+1.0* Bandwidth	Initial values
083–120	0.0*Delay+0.0*Loss+1.0* Bandwidth	Final values
121–122	0.5*Delay+0.5*Loss+0.0* Bandwidth	Initial values
123–160	0.5*Delay+0.5*Loss+0.0* Bandwidth	Final values
161–162	0.5*Delay+0.0*Loss+0.5* Bandwidth	Initial values
163–200	0.5*Delay+0.0*Loss+0.5* Bandwidth	Final values
201–202	0.0*Delay+0.5*Loss+0.5* Bandwidth	Initial values
203–240	0.0*Delay+0.5*Loss+0.5* Bandwidth	Final values
241–242	0 × 3Delay+0 × 3Loss+0.3* Bandwidth	Initial values
243–280	0 × 3Delay+0 × 3Loss+0.3* Bandwidth	Final values

Table 14
3 × 3 Network Deep Learning Cluster vs CPN-RNN.

Updates	CPN-RNN	QoS delay	QoS loss	QoS bandwidth
Initialization	0	8	6	7
CP 001–280	280	9	4	9

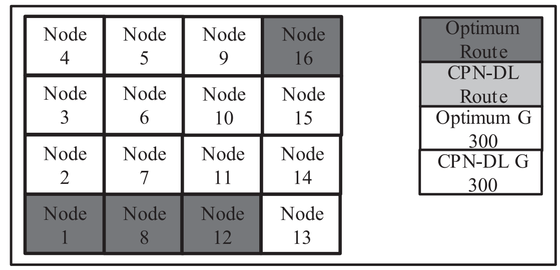


Fig. 11. 4 × 4 Network Final CPN-DL Route.

fundamental to take relevant optimum decisions. Three different strategic Cognitive Packets are chosen (CP 107, CP 228 and CP 341) for the 4 × 4 network validation, where each one has a different Goal (Table 15).

Table 13
4 × 4 Network Deep Learning Cluster.

Average	Cyber user	Cyber packet	Cyber node	QoS delay	QoS loss	QoS bandwidth
Error	6.96E−10	7.34E−10	9.93E−10	9.36E−10	9.23E−10	9.16E−10
Iterations	58.00	108.00	1017.87	145.29	148.50	133.88

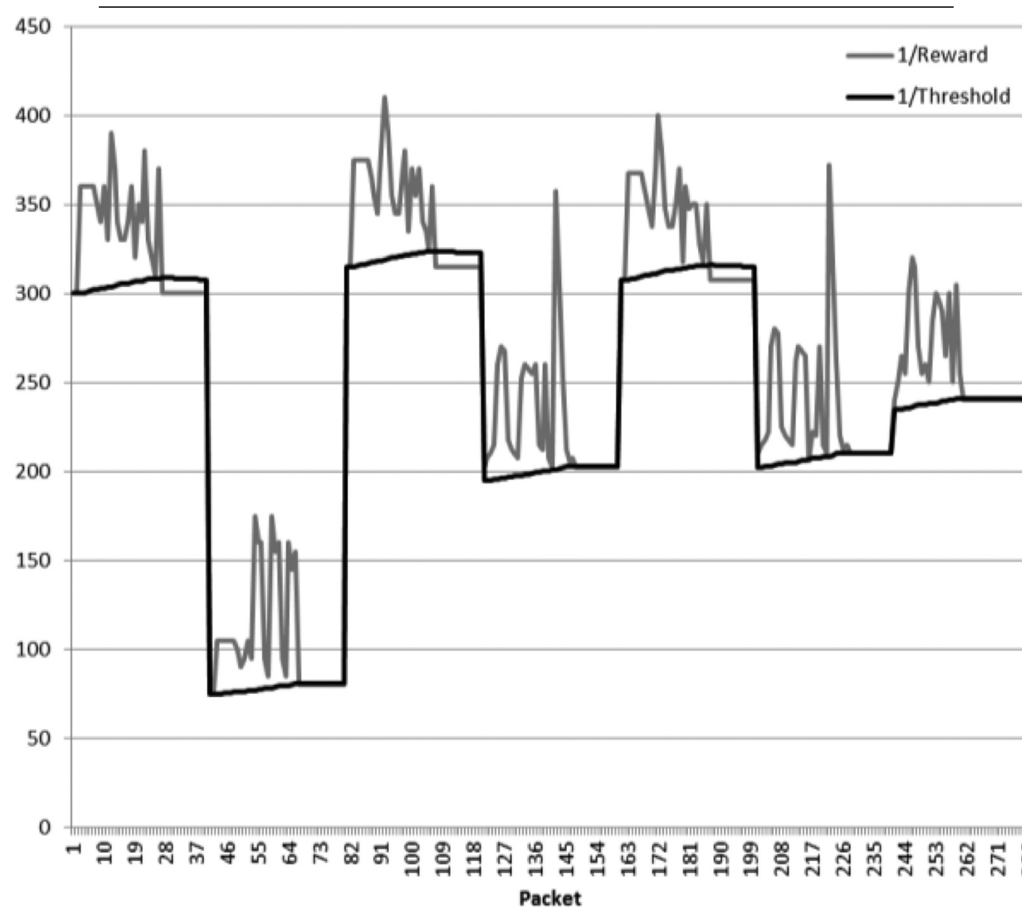


Fig. 10. 4 × 4 Network QoS DL Cluster validation.

Table 15
4×4 Network DL Management Cluster Validation.

Variable	Cognitive packet: 107 G:1.0*D + 0.0*L + 0.0*B		Cognitive packet: 228 G: 0.5*D + 0.5*L + 0.0*B		Cognitive packet: 341 G: 0.3*D + 0.3*L + 0.3*B	
	Δ=0.0	Δ=0.1	Δ=0.0	Δ=0.1	Δ=0.0	Δ=0.1
Cyber I_{cmc}	5E−11	3.4E−4	5E−11	3.4E−4	5E−11	3.4E−4
Cyber Y_{cmc}	0.9994	0.9969	0.9994	0.9969	0.9994	0.9969
QoS-Delay I_{qmc}	0.8000	0.8000	0.4000	0.4000	0.2666	0.2666
QoS-Loss I_{qmc}	0.0000	0.0000	0.2875	0.2875	0.1916	0.1916
QoS-Band I_{qmc}	0.0000	0.0000	0.0000	0.0000	0.2716	0.2716
QoS-Delay Y_{qmc}	0.1444	0.1444	0.2523	0.2523	0.3361	0.3361
QoS-Loss Y_{qmc}	0.9994	0.9994	0.3195	0.3195	0.4132	0.4132
QoS- Band Y_{qmc}	0.9994	0.9994	0.9994	0.9994	0.3319	0.3319
CEO I_{CEOmc}	0.1000	0.1000	0.1000	0.1000	0.9000	0.9000
CEO $w_{CEOmc}^{-}(c)$	0.0000	0.9999	0.0000	0.9999	0.0000	0.9999
CEO Y_{CEOmc}	0.9994	0.5746	0.9994	0.5746	0.9994	0.1305
Routing Decision	CPN Gate-6 Node 8	DL-Delay Gate-3 Node 5	CPN Gate-6 Node 8	DL-Delay Gate-6 Node 8	CPN Gate-6 Node 8	DL-Band Gate-3 Node 5

Table 16
5×5 Network QoS Deep Learning Cluster Validation – Simulation Parameters.

Cognitive packet	Goal	QoS metric
1500	Network Initialization Cognitive Packets	
01–02	1.0*Delay+0.0*Loss+0.0* Bandwidth	Initial values
03–50	1.0*Delay+0.0*Loss+0.0* Bandwidth	Final values

Table 17
5×5 Network Deep Learning Cluster.

Average	Cyber user	Cyber packet	Cyber node	QoS delay	QoS loss	QoS bandwidth
Error	7.56E−13	8.60E−13	9.91E−13	9.41E−13	9.30E−13	9.30E−13
Iterations	62	125	2128.68	221.11	182.40	200.71

Table 18
5×5 Network Deep Learning Cluster vs CPN-RNN.

Updates	CPN-RNN	QoS delay	QoS loss	QoS bandwidth
Initialization	0	8	20	7
CP 01–50	50	1	0	0

4.6. 5 × 5 node network – QoS DL cluster validation

The 5×5 network is simulated with a continuous 1550 Cognitive Packet stream. The first 1500 packets are used to initialize the CPN network with a single 1.0*Delay Goal after 50 packets whereas QoS metric changes 2 packets after the Goal is selected following T_l=0.999*T_{l-1} + 0.01*R (Table 16).

The Network keep sending Cognitive Packets until the value of the 1/Reward is lesser than the 1/Threshold. When the new best route is discovered; the CPN Threshold adapts gradually to the original value. The average Error and Iteration values for the different Deep Learning Clusters are represented on Table 17.

Table 18 shows the number of updates for the DL cluster and the CPN-RNN.

The CPN-RNN algorithm continuously updates its network weighs whereas the Delay QoS DL Cluster only updates its route once when a better route is found (Fig. 12).

4.7. 5 × 5 node network – DL management cluster validation

The results provided by the DL management cluster are shown on Table 19.

Fig. 13 shows the final CPN-DL route follows the Optimum Route in a 5×5 Network.

Table 19
5×5 Network DL Management Cluster Validation.

Variable	Cognitive packet: 34 G:1.0*D + 0.0*L + 0.0*B	
	Δ=0.0	Δ=0.1
Cyber I_{cmc}	5.14E−14	3.47E−04
Cyber Y_{cmc}	0.9994	0.9969
QoS-Delay I_{qmc}	0.5590	0.5590
QoS-Loss I_{qmc}	0.0000	0.0000
QoS-Band I_{qmc}	0.0000	0.0000
QoS-Delay Y_{qmc}	0.1945	0.1945
QoS-Loss Y_{qmc}	0.9994	0.9994
QoS-Band Y_{qmc}	0.9994	0.9994
CEO I_{CEOmc}	0.1000	0.1000
CEO $w_{CEOmc}^{-}(c)$	0.0000	0.9999
CEO Y_{CEOmc}	0.9994	0.5746
Routing Decision	CPN Gate-8 Node 10	DL-Delay Gate-4 Node 6

Table 20
6×6 Network QoS Deep Learning Cluster Validation – Simulation Parameters.

Cognitive packet	Goal	QoS metric
2500	Network Initialization Cognitive Packets	
01–02	1.0*Delay+0.0*Loss+0.0* Bandwidth	Initial values
03–60	1.0*Delay+0.0*Loss+0.0* Bandwidth	Final values

4.8. 6 × 6 node network – QoS DL cluster validation

The 6×6 network is simulated with a continuous 2560 Cognitive Packet stream. The first 2500 packets are used to initialize the CPN network with a single 1.0*Delay Goal after 60 packets whereas QoS metric changes 2 packets after the Goal is selected following T_l=0.99*T_{l-1} + 0.01*R (Table 20).

The number of Cognitive Packets sent to find the best route increases as the network size expands. The average Error and Iteration values for the different Deep Learning Clusters are represented on Table 21.

Table 22 shows the number of updates for the DL cluster and the CPN-RNN.

The CPN-RNN algorithm continuously updates its network weighs whereas on this evaluation; the Deep Learning Clusters have already learnt the best route during the network initialization stage (Fig. 14).

4.9. 6 × 6 node network – DL management cluster validation

The results provided by the DL management cluster are shown on Table 23.

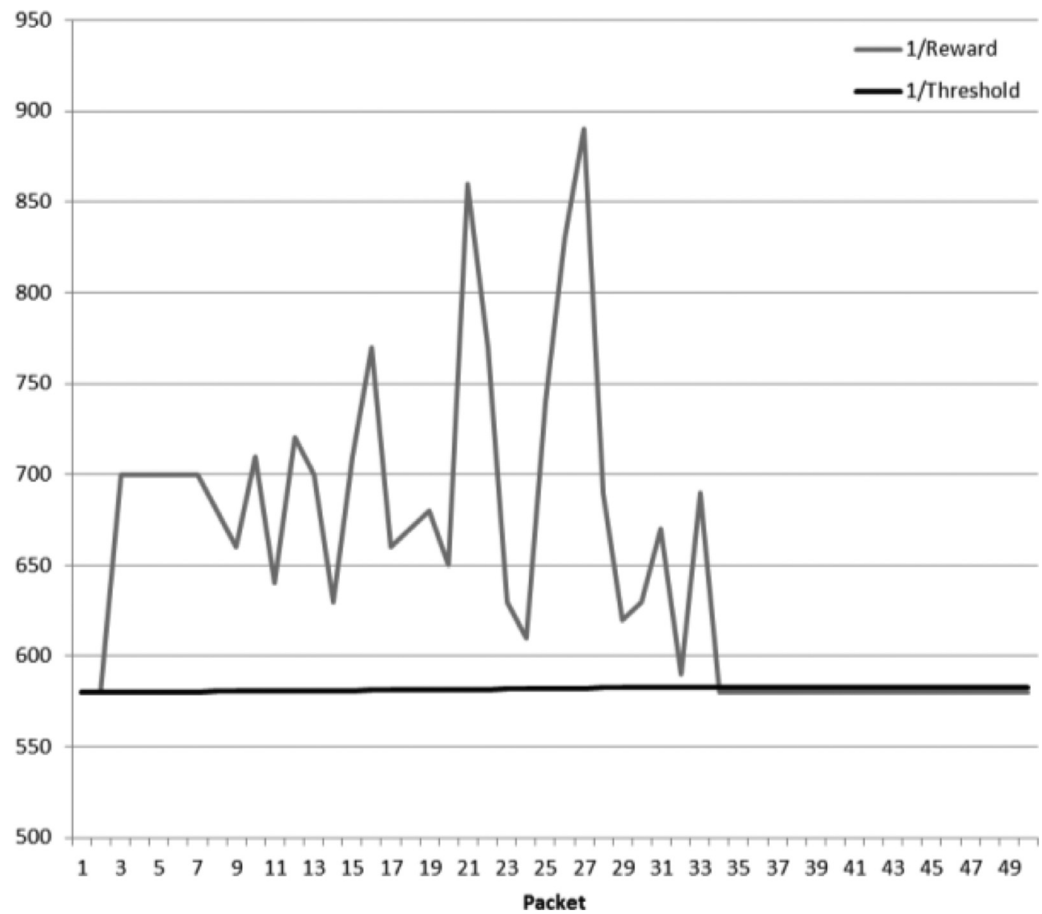


Fig. 12. 5×5 Network QoS DL Cluster validation.

Table 21
6×6 Network Deep Learning Cluster.

Average	Cyber user	Cyber packet	Cyber node	QoS delay	QoS loss	QoS bandwidth
Error	7.56E−13	8.60E−13	9.92E−13	9.53E−13	9.05E−13	9.44E−13
Iterations	62	125	1962.36	213.42	193.33	185.27

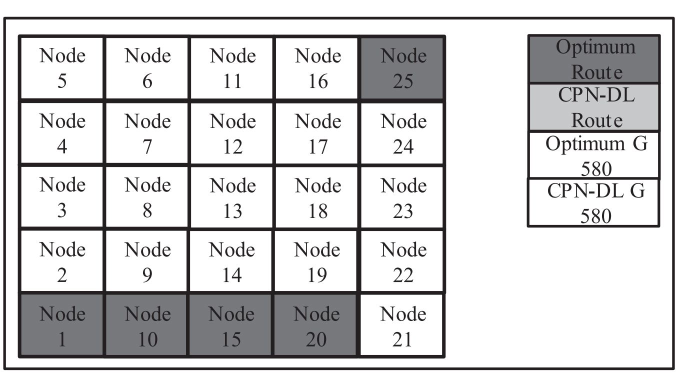


Fig. 13. 5×5 Network Final CPN-DL Route.

Table 22
6×6 Network Deep Learning Cluster vs CPN-RNN.

Updates	CPN-RNN	QoS delay	QoS loss	QoS bandwidth
Initialization	0	12	3	11
CP 01–60	60	0	0	0

Table 23
6×6 Network DL Management Cluster Validation.

Variable	Cognitive packet: 51 G:$1.0^{*}D+0.0^{*}L+0.0^{*}B$	
	Δ=0.0	Δ=0.1
Cyber I_{cmc}	5.14E−14	3.62E−04
Cyber Y_{cmc}	0.9994	0.9968
QoS-Delay I_{qmc}	0.6010	0.6010
QoS-Loss I_{qmc}	0.0000	0.0000
QoS-Band I_{qmc}	0.0000	0.0000
QoS-Delay Y_{qmc}	0.1834	0.1834
QoS-Loss Y_{qmc}	0.9994	0.9994
QoS- Band Y_{qmc}	0.9994	0.9994
CEO I_{CEOmc}	0.1000	0.1000
CEO $w_{CEOmc}^{-}(c)$	0.0000	0.9999
CEO Y_{CEOmc}	0.9994	0.5746
Routing Decision	CPN Gate-10 Node 12	DL-Delay Gate-5 Node 7

Fig. 15 shows the final CPN-DL and the Optimum Route in a 6×6 Network. There are only two nodes of difference between the two routes.

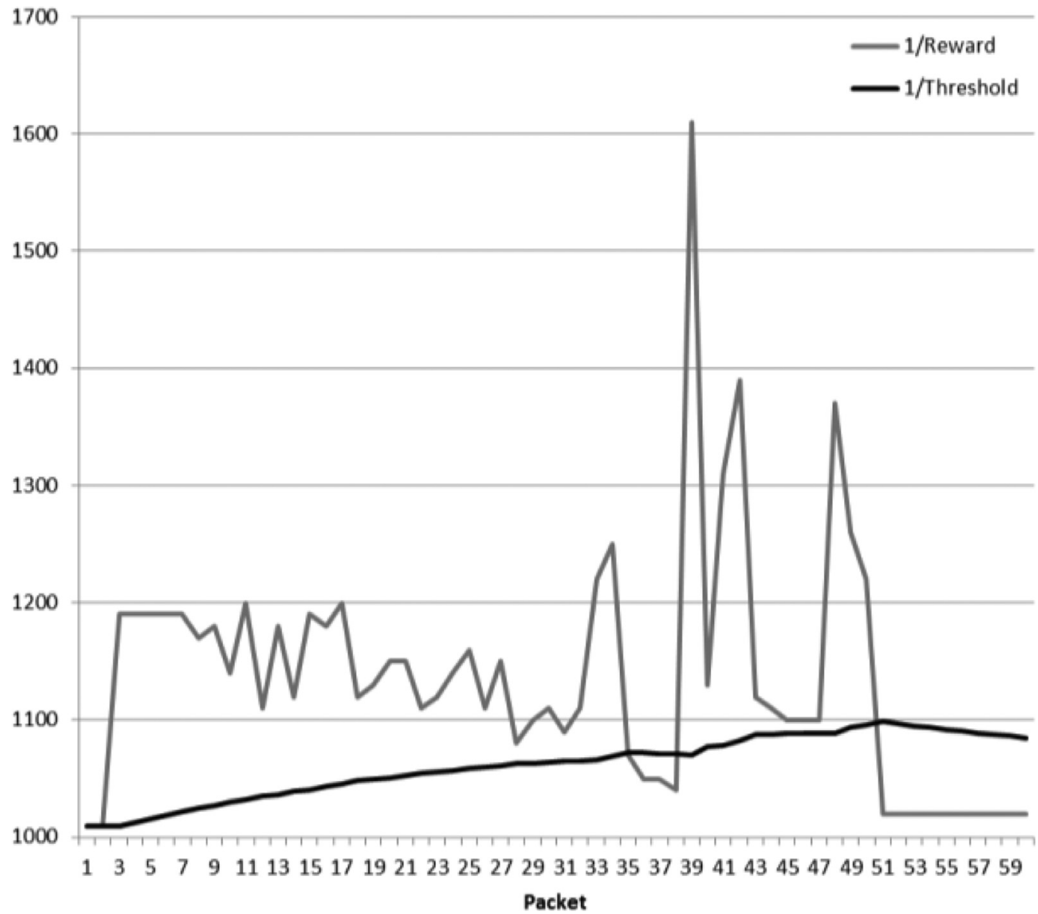


Fig. 14. 6×6 Network QoS DL Cluster validation.

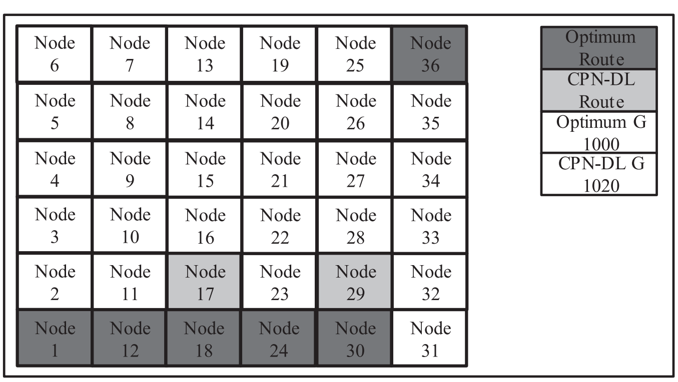


Fig. 15. 6×6 Network Final CPN-DL Route.

Table 24
7×7 Network QoS Deep Learning Cluster Validation – Simulation Parameters.

Cognitive Packet	Goal	QoS metric
3500	Network Initialization Cognitive Packets	
01–02	1.0*Delay+0.0*Loss+0.0* Bandwidth	Initial values
03–60	1.0*Delay+0.0*Loss+0.0* Bandwidth	Final values

4.10. 7×7 node network – QoS DL cluster validation

The 7×7 network is simulated with a continuous 3560 Cognitive Packet stream. The first 3500 packets are used to initialize the CPN network with a single 1.0*Delay Goal after 60 packets whereas QoS metric changes 2 packets after the Goal is selected following T_l=0.999*$T_{l\text{-}1}$ + 0.01*R (Table 24).

The Network does not converge to the optimum Goal value due the node threshold updates. The average Error and Iteration values for the different Deep Learning Clusters are represented on Table 25.

Table 26 shows the number of updates for the DL cluster and the CPN-RNN.

The Delay and Bandwidth QoS DL clusters update their route following the QoS network changes (Fig. 16).

4.11. 7×7 node network – DL management cluster validation

The results provided by the DL management cluster are shown on Tables 27.

Fig. 17 shows the final CPN-DL route and the Optimum Route in a 7×7 Network. The difference between paths has widens due the network size has increased.

4.12. 8×8 node network – QoS DL cluster validation

The 8×8 network is simulated with a continuous 4575 Cognitive Packet stream. The first 4500 packets are used to initialize the CPN network with a single 1.0*Delay Goal after 75 packets whereas QoS metric changes 2 packets after the Goal is selected following T_l=0.999*$T_{l\text{-}1}$ + 0.01*R (Table 28).

The Reward obtained by the Cognitive Packets follows a downward trend until the network converges with some spikes due the final path adaptation from independent node layers. The average Error and Iteration values for the different Deep Learning Clusters are represented on Table 29.

Table 30 shows the number of updates for the DL cluster and the CPN-RNN.

Table 25
7 × 7 Network Deep Learning Cluster.

Average	Cyber user	Cyber packet	Cyber node	QoS delay	QoS loss	QoS bandwidth
Error	7.56E−13	8.60E−13	9.91E−13	9.39E−13	9.24E−13	9.05E−13
Iterations	62	125	1821.98	201.37	185.14	176

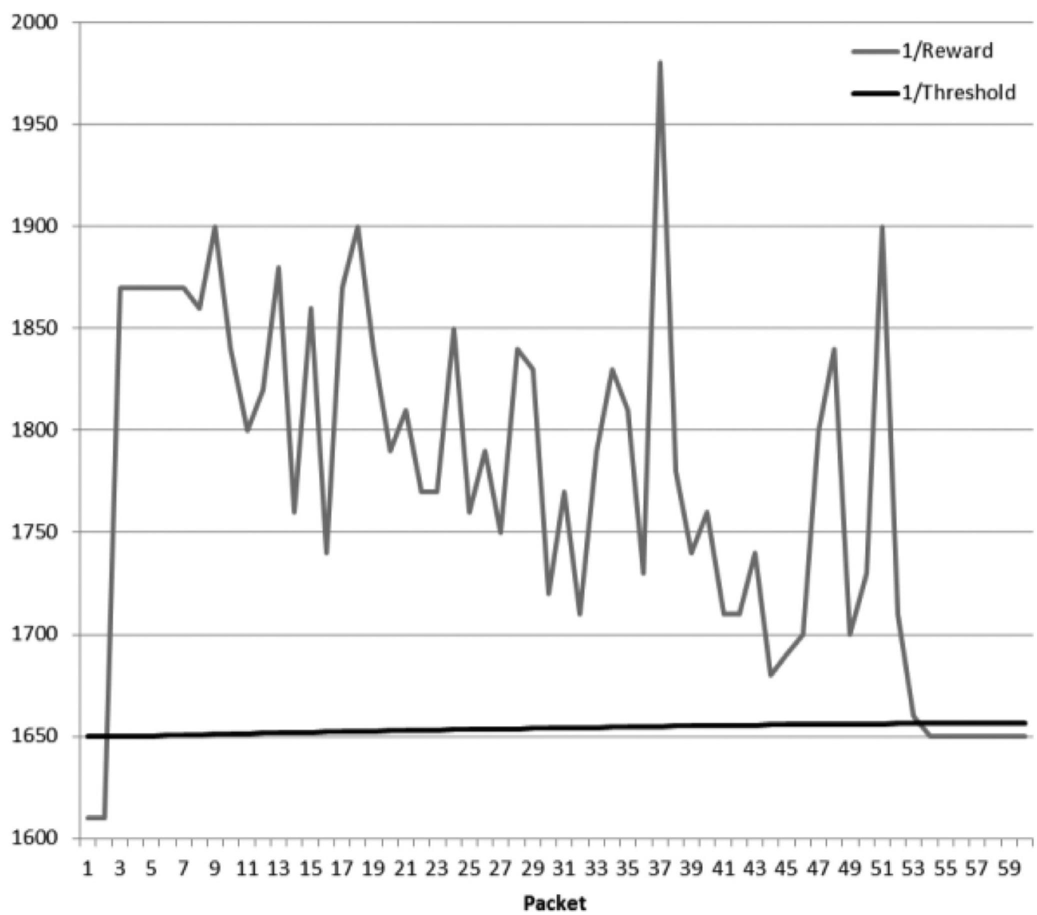


Fig. 16. 7 × 7 Network QoS DL Cluster validation.

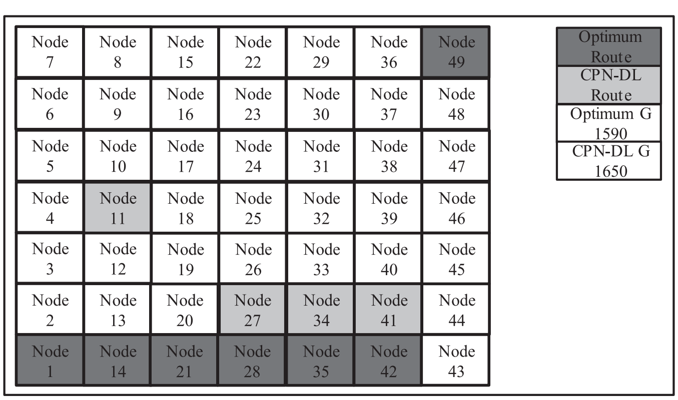


Fig. 17. 7 × 7 Network Final CPN-DL Route.

Table 26
7 × 7 Network Deep Learning Cluster vs CPN-RNN.

Updates	CPN-RNN	QoS delay	QoS loss	QoS bandwidth
Initialization	0	18	7	14
CP 01–60	60	1	0	2

The Bandwidth QoS DL Cluster updates its route following the QoS network changes (Fig. 18).

4.13. 8 × 8 node network – DL management cluster validation

The results provided by the DL management cluster are shown on Table 29.

Fig. 19 shows the final CPN-DL route and the Optimum Route in a 8 × 8 Network. The CPN-DL route is close to the optimum route.

Table 29
8 × 8 Network Deep Learning Cluster.

Average	Cyber user	Cyber packet	Cyber node	QoS delay	QoS loss	QoS bandwidth
Error	7.56E−13	8.60E−13	9.91E−13	9.21E−13	8.82E−13	8.72E−13
Iterations	62	125	1717.36	169.50	158.00	169.56

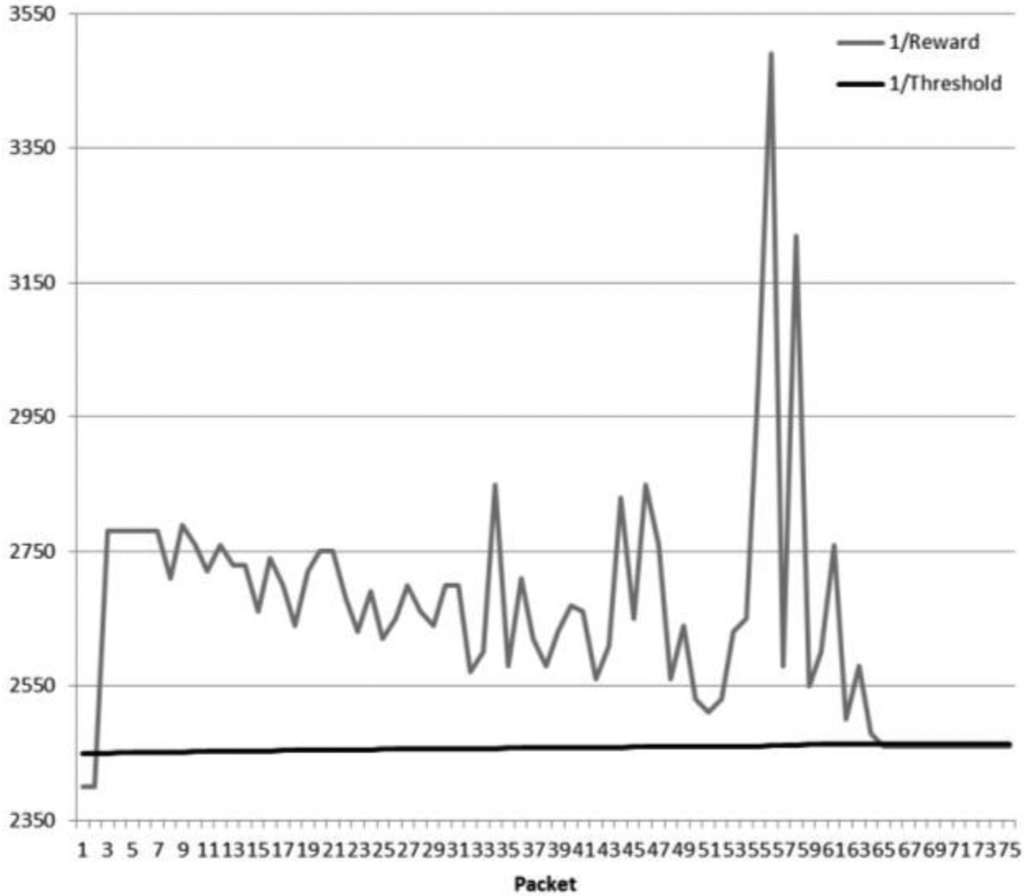


Fig. 18. 8 × 8 Network QoS DL Cluster validation.

Table 27
7 × 7 Network DL Management Cluster Validation.

Variable	Cognitive packet: 54 G:1.0*D + 0.0*L + 0.0*B	
	Δ=0.0	Δ=0.1
Cyber I_{cmc}	5.14E−14	3.62E−04
Cyber Y_{cmc}	0.9994	0.9968
QoS-Delay I_{qmc}	0.6630	0.6630
QoS-Loss I_{qmc}	0.0000	0.0000
QoS-Band I_{qmc}	0.0000	0.0000
QoS-Delay Y_{qmc}	0.1692	0.1692
QoS-Loss Y_{qmc}	0.9994	0.9994
QoS- Band Y_{qmc}	0.9994	0.9994
CEO I_{CEOmc}	0.1000	0.1000
CEO $w_{CEOmc}^{-}(c)$	0.0000	0.9999
CEO Y_{CEOmc}	0.9994	0.5746
Routing Decision	CPN Gate-09 Node 12	DL-Delay Gate-6 Node 8

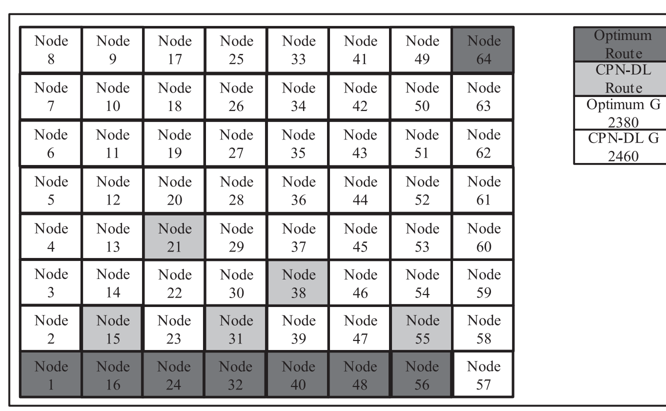


Fig. 19. 8 × 8 Network Final CPN-DL Route.

Table 28
8 × 8 Network QoS Deep Learning Cluster Validation – Simulation Parameters.

Cognitive packet	Goal	QoS metric
4500	Network Initialization Cognitive Packets	
01–02	1.0*Delay+0.0*Loss+0.0* Bandwidth	Initial values
03–75	1.0*Delay+0.0*Loss+0.0* Bandwidth	Final values

Table 30
8 × 8 Network Deep Learning Cluster vs CPN-RNN.

Updates	CPN-RNN	QoS delay	QoS loss	QoS bandwidth
Initialization	0	10	7	8
CP 01–75	75	0	0	1

4.14. 9 × 9 node network – QoS DL cluster validation

The 9 × 9 network is simulated with a continuous 4585 Cognitive Packet stream. The first 4500 packets are used to initialize the CPN network with a single 1.0*Delay Goal after 85 packets whereas QoS metric changes 2 packets after the Goal is selected following T_l=0.999*$T_{l\text{-}1}$ + 0.01*R (Tables 31 and 32).

The network increasingly searches the path that optimizes its Reward; even after finding a route that meets the threshold limit. The average Error and Iteration values for the different Deep Learning Clusters are represented on Table 33.

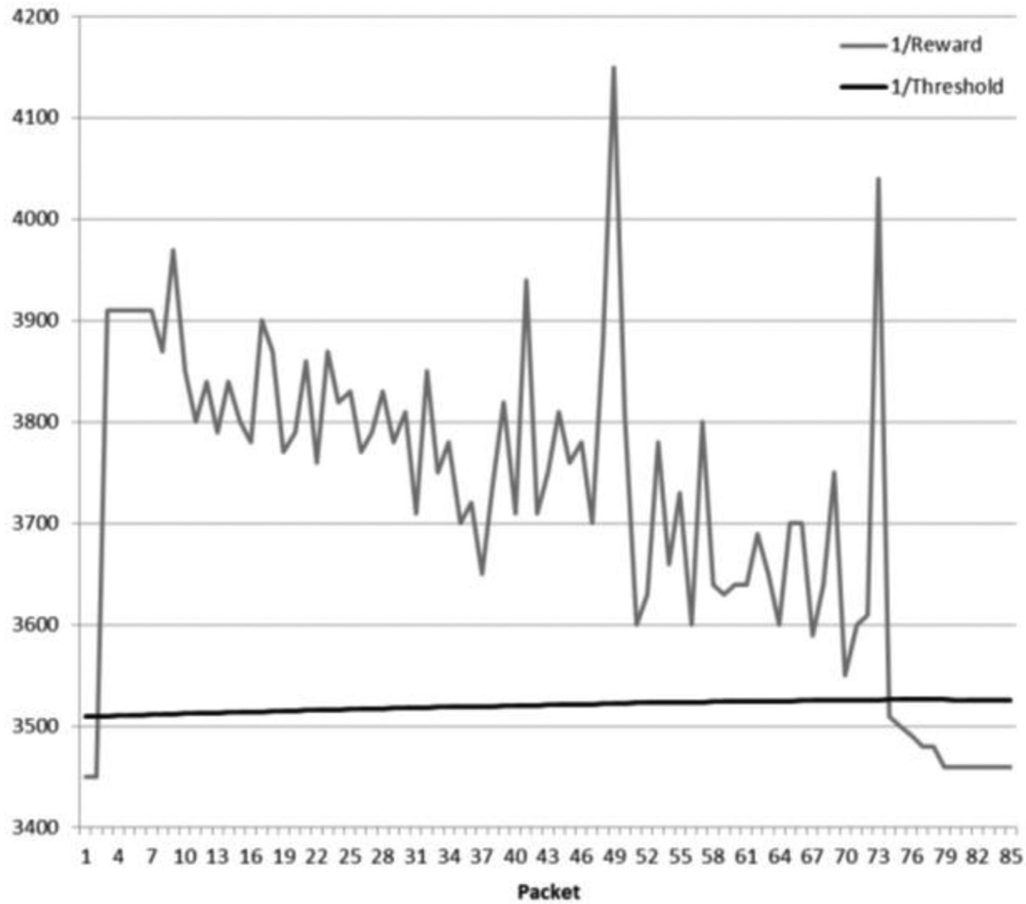


Fig. 20. 9×9 Network QoS DL Cluster validation.

Table 31
8×8 Network DL Management Cluster Validation.

Variable	Cognitive packet: 65 G:1.0*D+0.0*L+0.0*B	
	Δ=0.0	Δ=0.1
Cyber I_{cmc}	5.14E−14	3.62E−04
Cyber Y_{cmc}	0.9994	0.9968
QoS-Delay I_{qmc}	0.7450	0.7450
QoS-Loss I_{qmc}	0.0000	0.0000
QoS-Band I_{qmc}	0.0000	0.0000
QoS-Delay Y_{qmc}	0.1534	0.1534
QoS-Loss Y_{qmc}	0.9994	0.9994
QoS- Band Y_{qmc}	0.9994	0.9994
CEO I_{CEOmc}	0.1000	0.1000
CEO $w_{CEOmc}^{-}(c)$	0.0000	0.9999
CEO Y_{CEOmc}	0.9994	0.5746
Routing Decision	CPN Gate-13 Node 15	DL-Delay Gate-8 Node 10

Table 32
9×9 Network QoS Deep Learning Cluster Validation – Simulation Parameters.

Cognitive packet	Goal	QoS metric
4500	Network Initialization Cognitive Packets	
01–02	1.0*Delay+0.0*Loss+0.0* Bandwidth	Initial values
03–85	1.0*Delay+0.0*Loss+0.0* Bandwidth	Final values

Table 34
9×9 Network Deep Learning Cluster vs CPN-RNN.

Updates	CPN-RNN	QoS delay	QoS loss	QoS bandwidth
Initialization	0	6	8	5
CP 01–85	85	1	0	2

Table 34 shows the number of updates for the DL cluster and the CPN-RNN.

The Delay and Bandwidth QoS DL Clusters update their route following the QoS network changes (Fig. 20).

4.15. 9×9 node network – DL management cluster validation

The results provided by the DL management are shown on Table 35.

Fig. 21 shows the final CPN-DL route and the Optimum Route in a 9×9 Network. The different between routes does not widen as the network increases due the decentralized routing protocols where network nodes are independent.

Table 33
9×9 Network Deep Learning Cluster.

Average	Cyber user	Cyber packet	Cyber node	QoS delay	QoS loss	QoS bandwidth
Error	7.56E−13	8.60E−13	9.92E−13	9.32E−13	9.40E−13	9.45E−13
Iterations	62	125	1874.09	204.57	172.25	170.71

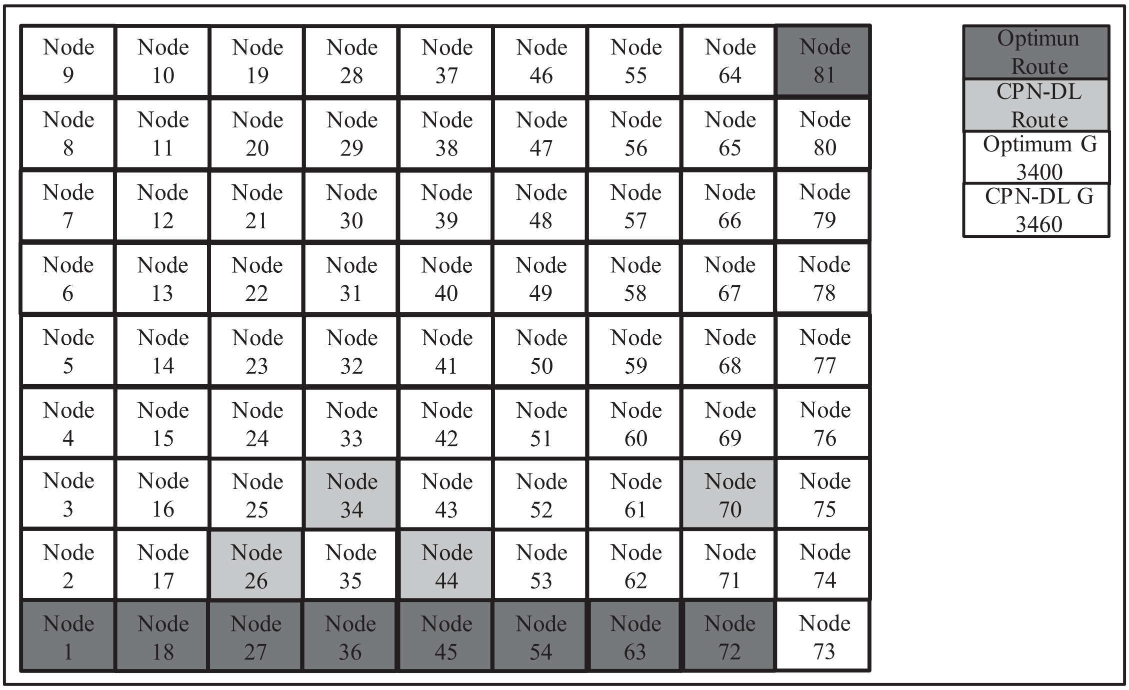


Fig. 21. 9 × 9 Network Final CPN-DL Route.

Table 35
9 × 9 Network DL Management Cluster Validation.

Variable	Cognitive packet: 74 G:1.0*D + 0.0*L + 0.0*B	
	Δ=0.0	Δ=0.1
Cyber I_{cmc}	5.14E− 14	3.62E−04
Cyber Y_{cmc}	0.9994	0.9968
QoS-Delay I_{qmc}	0.8510	0.8510
QoS-Loss I_{qmc}	0.0000	0.0000
QoS-Band I_{qmc}	0.0000	0.0000
QoS-Delay Y_{qmc}	0.1369	0.1369
QoS-Loss Y_{qmc}	0.9994	0.9994
QoS- Band Y_{qmc}	0.9994	0.9994
CEO I_{CEOmc}	0.1000	0.1000
CEO $w_{CEOmc}^{-}(c)$	0.0000	0.9999
CEO Y_{CEOmc}	0.9994	0.5746
Routing Decision	CPN Gate-16 Node 18	DL-Delay Gate-10 Node 12

Table 36
10 × 10 Network QoS Deep Learning Cluster Validation – Simulation Parameters.

Cognitive packet	Goal	QoS metric
4500	Network Initialization Cognitive Packets	
001–002	1.0*Delay+0.0*Loss+0.0* Bandwidth	Initial values
003–100	1.0*Delay+0.0*Loss+0.0* Bandwidth	Final values

4.16. 10 × 10 node network – QoS DL cluster validation

The 10 × 10 network is simulated with a continuous 4600 Cognitive Packet stream. The first 4500 packets are used to initialize the CPN network with a single 1.0*Delay Goal after 100 packets whereas QoS metric changes 2 packets after the Goal is selected following $T_l = 0.999{*}T_{l\text{-}1} + 0.01{*}R$ (Table 36).

The Reward follows spikes even after meeting the threshold level due the adaptation of different network layers. The average Error and Iteration values for the different Deep Learning Clusters are represented on Table 37.

Table 37
10 × 10 Network Deep Learning Cluster.

Average	Cyber user	Cyber packet	Cyber node	QoS delay	QoS loss	QoS bandwidth
Error	7.56E−13	8.60E−13	9.91E−13	1.64E−12	9.10E−13	1.39E−12
Iterations	62	125	1610.43	216	163.25	181.67

Table 38
10 × 10 Network Deep Learning Cluster vs CPN-RNN.

Updates	CPN-RNN	QoS delay	QoS loss	QoS bandwidth
Initialization	0	13	8	11
CP 001–100	85	1	0	1

Table 38 shows the number of updates for the DL cluster and the CPN-RNN.

Similar to previous network sizes; the Delay and Bandwidth QoS DL clusters update their route following the QoS network changes (Fig. 22).

4.17. 10 × 10 node network – DL management cluster validation

The results provided by the DL management cluster are shown on Table 39.

Fig. 23 shows the final CPN-DL route and the Optimum Route in a 10 × 10 Network. The CPN-DL Route almost follows the optimum route.

4.18. General n × n network validation results

The 3 × 3, 4 × 4 and 5 × 5 node network routing decisions for the DL clusters are shown on Table 40. CP is the number of Cognitive Packets the DL clusters need to adapt to the new route and G is the final Goal.

The number of Cognitive Packets required to find the optimum route increases as the network expands. The network adaptation between different QoS metrics is consistent.

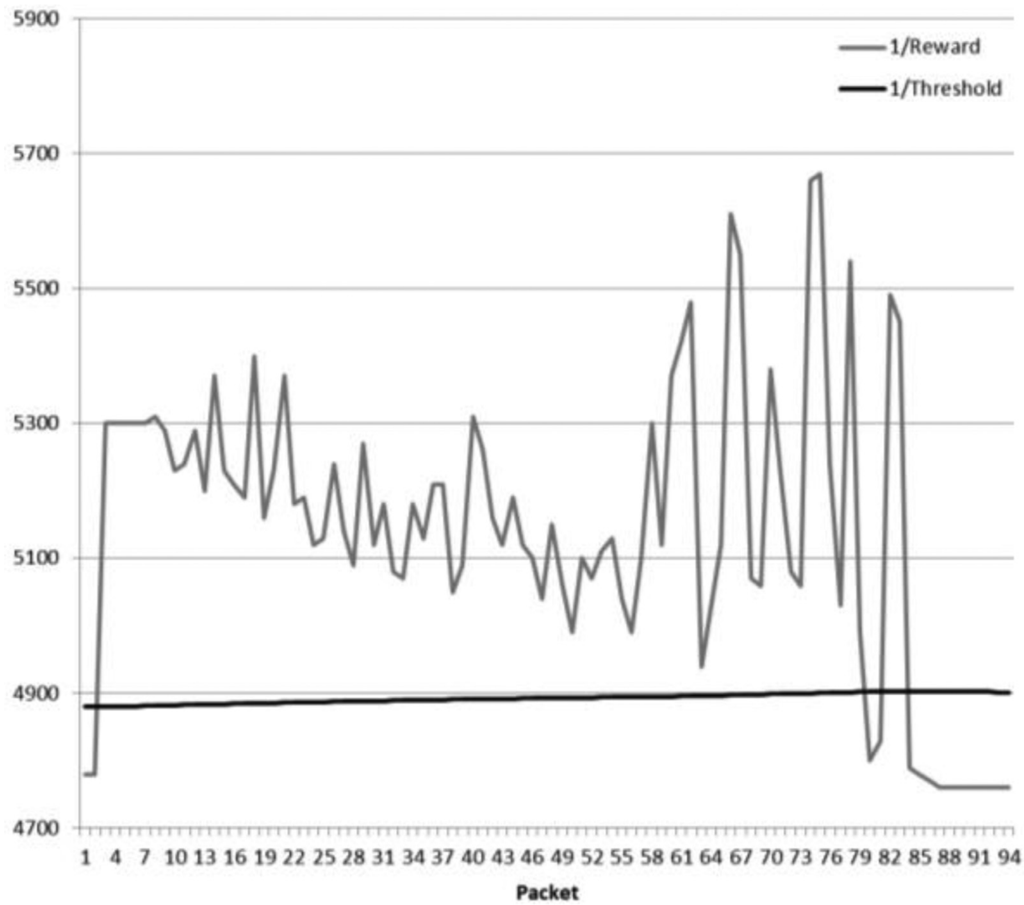


Fig. 22. 10 × 10 Network QoS DL Cluster validation.

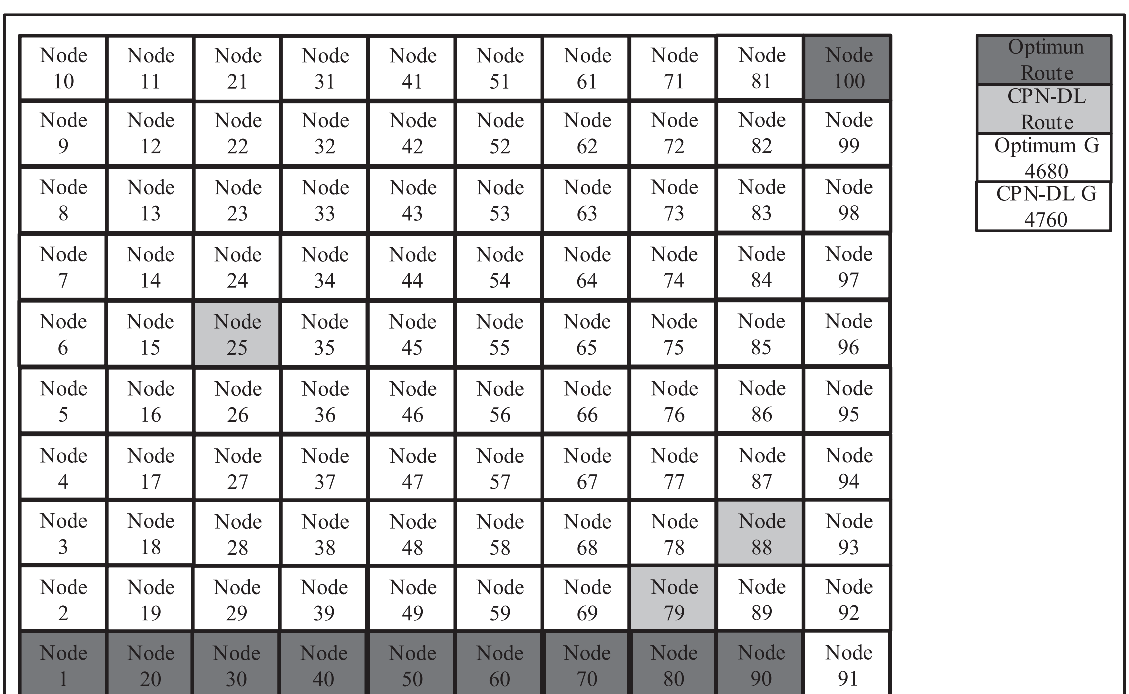


Fig. 23. 10 × 10 Network Final CPN-DL Route.

Table 39
10 × 10 Network DL Management Cluster Validation.

Variable	Cognitive packet: 80 G:1.0*D + 0.0*L + 0.0*B	
	Δ=0.0	Δ=0.1
Cyber I_{cmc}	5.14E−14	3.62E−04
Cyber Y_{cmc}	0.9994	0.9968
QoS-Delay I_{qmc}	0.9800	0.9800
QoS-Loss I_{qmc}	0.0000	0.0000
QoS-Band I_{qmc}	0.0000	0.0000
QoS-Delay Y_{qmc}	0.1211	0.1211
QoS-Loss Y_{qmc}	0.9994	0.9994
QoS- Band Y_{qmc}	0.9994	0.9994
CEO I_{CEOmc}	0.1000	0.1000
CEO $w_{CEOmc}^{-}(c)$	0.0000	0.9999
CEO Y_{CEOmc}	0.9994	0.5746
Routing	CPN Gate-18	DL-Delay Gate-9
Decision	Node 20	Node 11

Table 40
QoS DL Cluster Validation – Overall Results.

QoS	3 × 3 Network	4 × 4 Network	5 × 5 Network
1.0*D + 0.0*L + 0.0*B	Goal: 130.0 - CP: 8.0	Goal: 300.0 - CP: 25.0	Goal: 580.0 - CP: 31.0
0.0*D + 1.0*L + 0.0*B	Goal: 25.0 - CP: 4.0	Goal: 75.0 - CP: 26.0	Goal: 170.0 - CP: 40.0
0.0*D + 0.0*L + 1.0*B	Goal: 140.0 - CP: 7.0	Goal: 315.0 - CP: 27.0	Goal: 600.0 - CP: 31.0
0.5*D + 0.5*L + 0.0*B	Goal: 140.0 - CP: 7.0	Goal: 315.0 - CP: 25.0	Goal: 405.0 - CP: 25.0
0.5*D + 0.0*L + 0.5*B	Goal: 82.5 - CP: 3.0	Goal: 202.5 - CP: 26.0	Goal: 590.5 - CP: 31.0
0.0*D + 0.5*L + 0.5*B	Goal: 135.0 - CP: 7.0	Goal: 307.5 - CP: 25.0	Goal: 415.5 - CP: 25.0
0.3*D + 0.3*L + 0.3*B	Goal: 87.5 - CP: 6.0	Goal: 210.0 - CP: 26.0	Goal: 475.0 - CP: 32.0

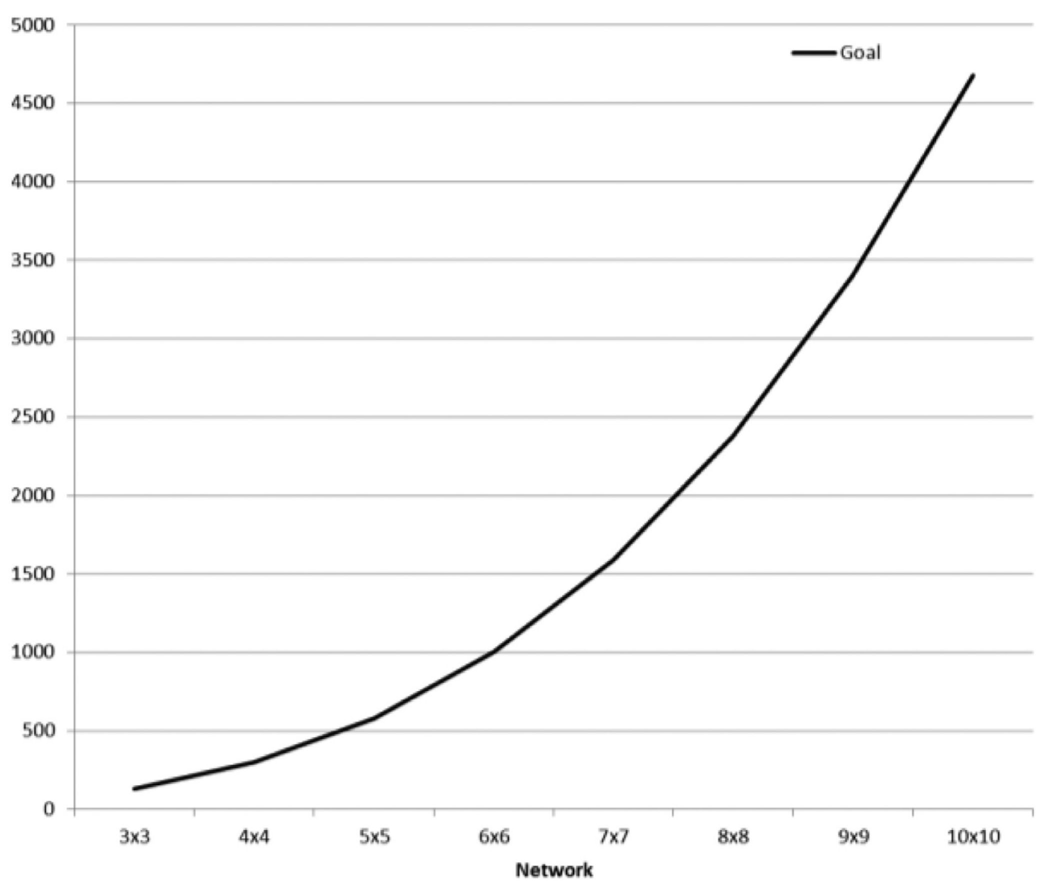


Fig. 24. $n \times n$ network goal.

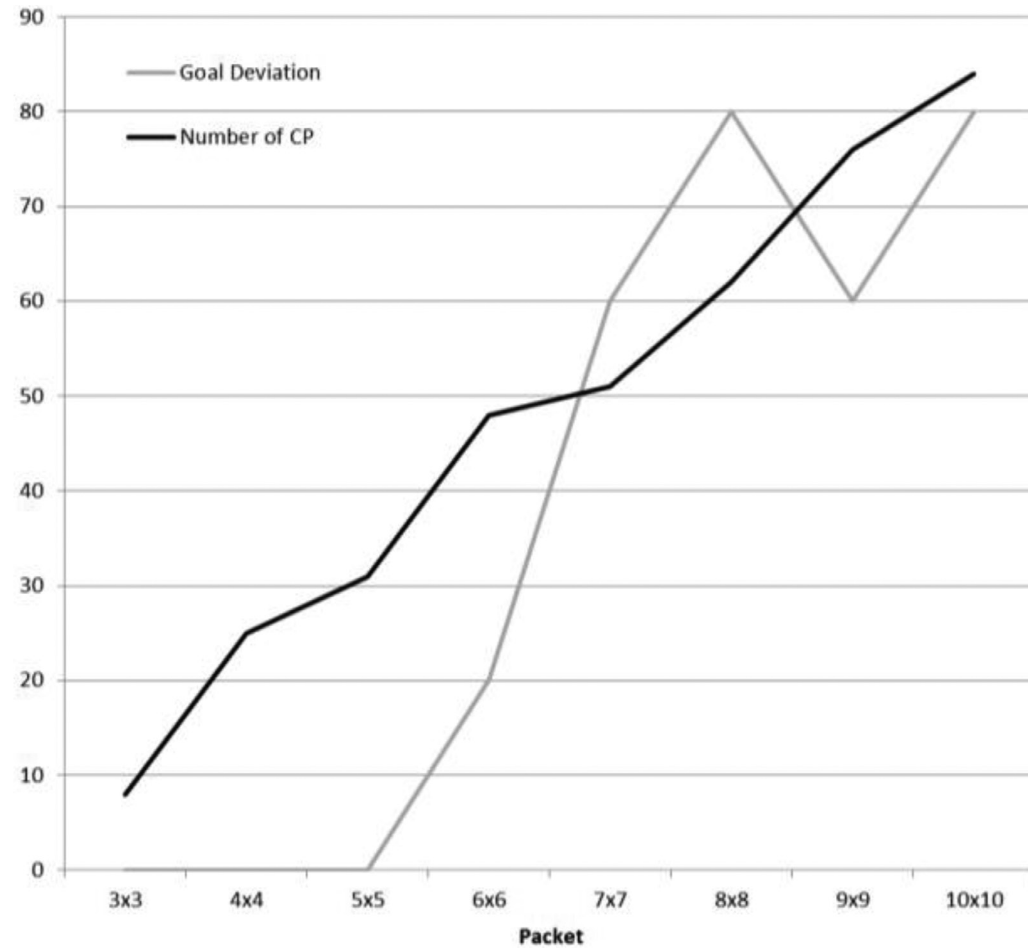


Fig. 25. $n \times n$ network goal deviation and number of Cognitive Packets.

Table 41
QoS DL Cluster Validation – General Results.

Network	QoS: $1.0^*D + 0.0^*L + 0.0^*B$				
	Initial goal	Final goal	Optimum goal	Deviation	CP packets
3×3	130	130	130	0.0	8
4×4	300	300	300	0.0	25
5×5	580	580	580	0.0	31
6×6	1010	1020	1000	20.0	48
7×7	1610	1650	1590	60.0	51
8×8	2400	2460	2380	80.0	62
9×9	3450	3460	3400	60.0	76
10×10	4780	4760	4680	80.0	84

The overall Quality of Service for 1.0*Delay routing decisions for the DL clusters are shown on Table 41 with the best Goal obtained by the CPN, the final Goal achieved by the CPN-DL after the QoS change and the optimum Goal. In addition; the Deviation between Final and Optimum Goal along the required number of Cognitive Packets during the QoS adaptation are also shown.

As the network grows; the Goal therefore also increases along with the required number of Cognitive Packets to find the optimum route as shown on Fig. 24.

Deviation was also expected to increase linearly; however due to the network decentralization routing algorithm, where decision layers are independent, its value remains constant against network sizes as represented on Fig. 25. It is expected Deviation will converge as the number of nodes increases.

4.19. Node threshold validation

This section analyses the impact of the parameter α when calculating the value of the node decision threshold $T_l = \alpha T_{l-1} + (1-\alpha)R_l$. The validation analyses the CPN-DL network stability and adaptability with different values assigned to the parameter α (0.9, 0.99, 0.999 and 0.9999) in the 6×6 network for the first 100 Cognitive Packets.

With $\alpha = 0.9$, the Network does not converge to the initial Reward (1010), this leads to Network Depression as it has already learnt there is a better route (1010) although the final path selected is the one that meets the Threshold level (1120). The convergence is quite fast; with only 31 Cognitive Packets (Fig. 26).

A larger value of $\alpha = 0.99$ makes the network to converge to a better Goal (1020), although not the optimum Goal value (1010), however an increased number of 48 Cognitive Packets is required; this is reflected as the network anxiety to find a stable route (Fig. 27).

The network does not converge to the optimum Goal (1010) with $\alpha = 0.999$; instead it requires more Cognitive Packets (81) increasing the network anxiety to find the final Goal route (1020) as shown on Fig. 28.

The network becomes instable with $\alpha = 0.9999$; it does not converge with 100 Cognitive Packets (Fig. 29).

We have analyzed the relation between the optimum Goal route and the required number of Cognitive Packets to find it.

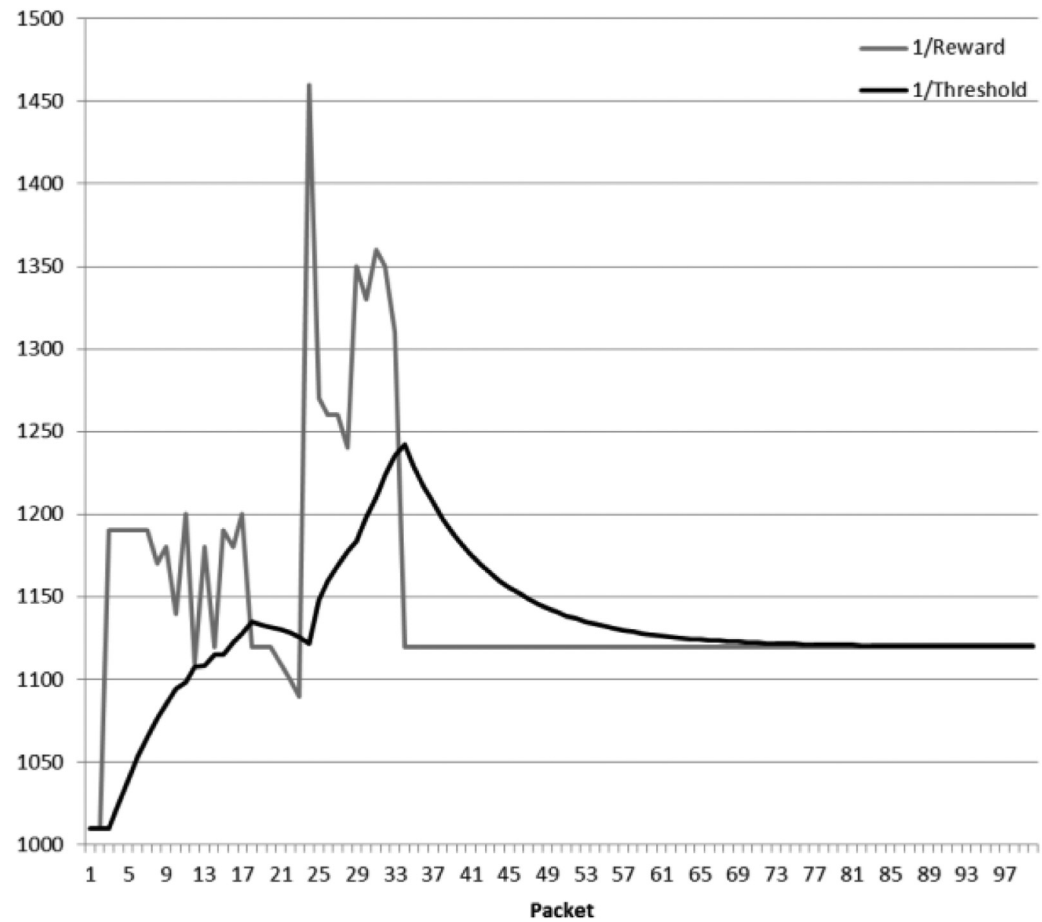


Fig. 26. 6×6 Network with $\alpha = 0.9$.

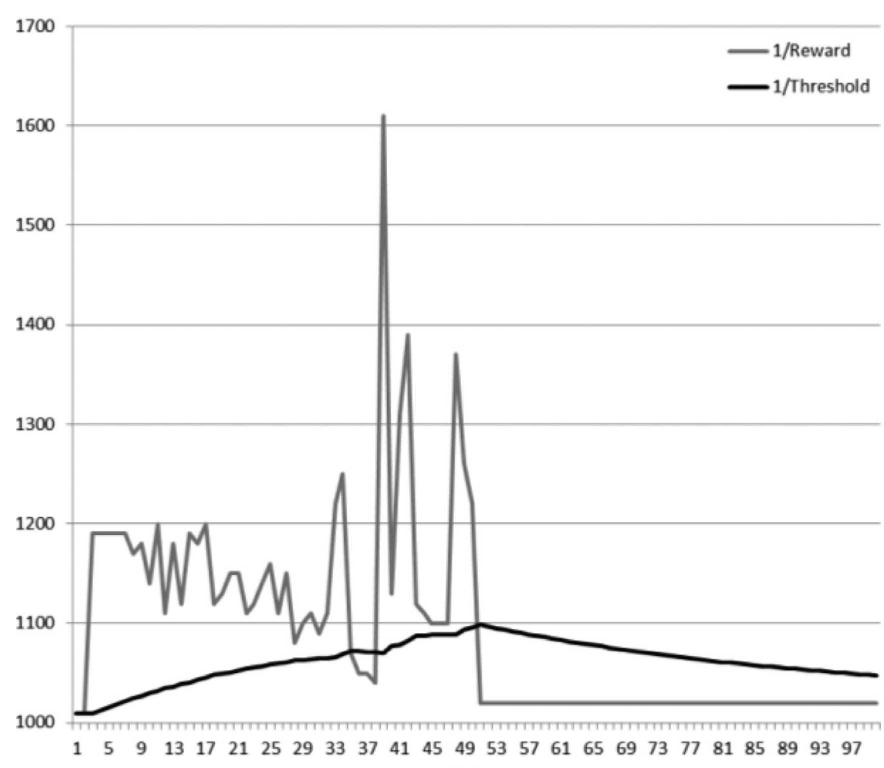


Fig. 27. 6×6 Network with $\alpha = 0.99$.

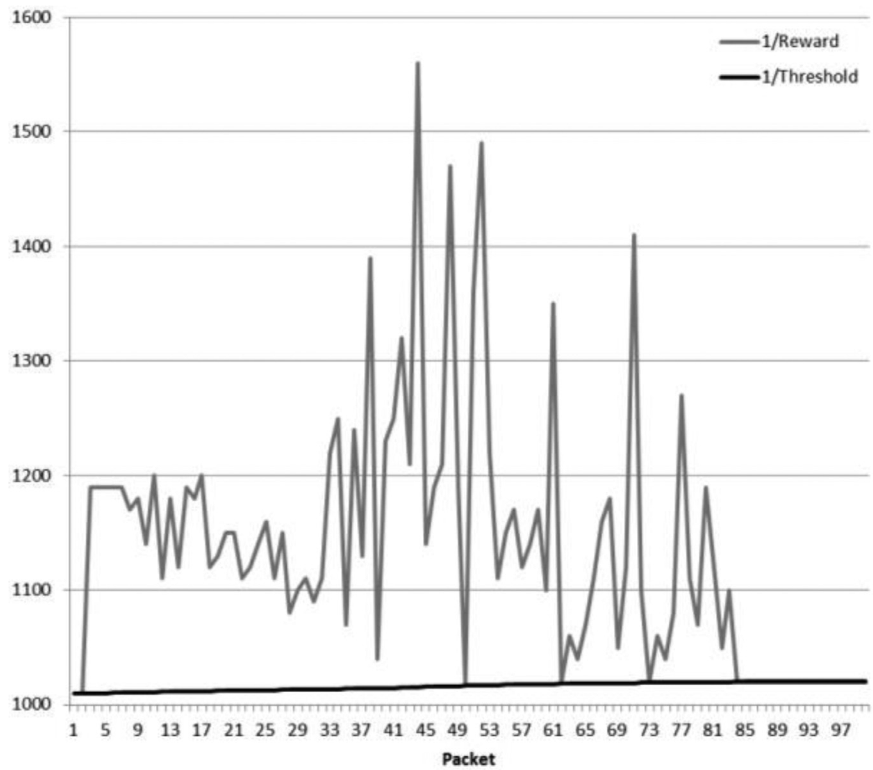


Fig. 28. 6×6 Network with $\alpha = 0.999$.

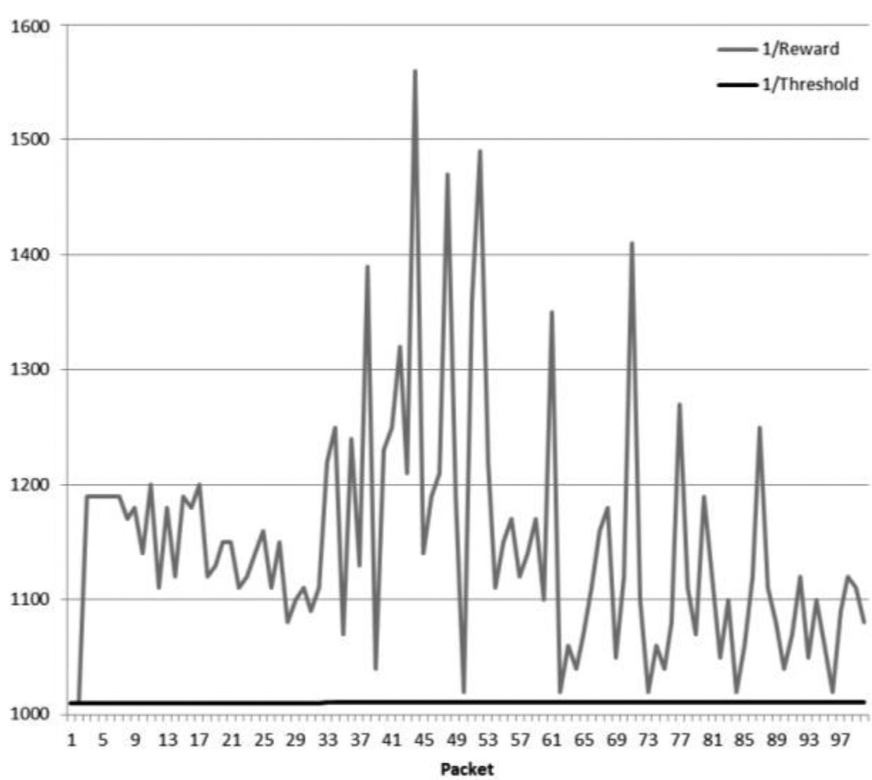


Fig. 29. 6×6 Network with $\alpha = 0.9999$.

Table 42
Node Threshold Validation – Overall Results.

α	Initial goal	Final goal	Cognitive packets	CPN status
0.9	1010	1120	31	Mayor depressed
0.99	1010	1020	48	Minor depressed
0.999	1010	1020	81	Minor anxiety
0.9999	1010	Unstable	Unstable	Mayor anxiety

Similar to human behaviour; the anxiety produced due to the route finding is in conflict with the depression due to not finding the optimum route (Table 42).

5. Conclusions

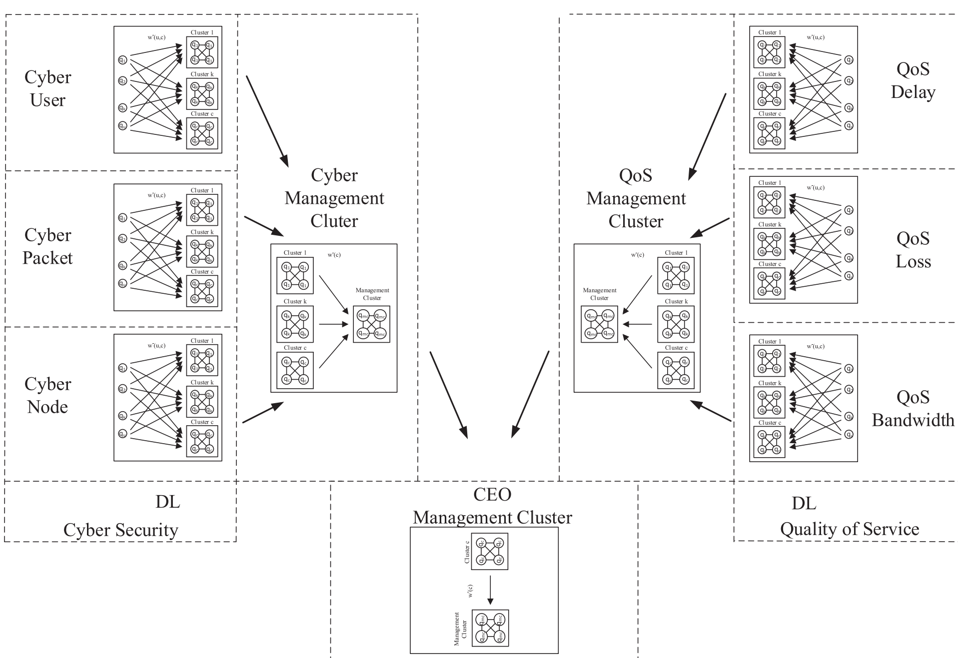
This paper has presented a biological inspired learning algorithm: the Random Neural Network with a Deep Learning Cluster structure. The CPN-RNN algorithm adapts very fast to variable QoS changes with fast decisions in short term memory; whereas the Deep Learning algorithm is slow to adapt to QoS changes as it learns from the CPN algorithm and stores routing information in long term memory. The CEO management cluster takes the right routing decisions based on the inputs from the QoS and Cyber DL management clusters. This allows the CPN to use a safe route in case of Cyber attack, or a QoS efficient route under normal conditions. The model has been validated it using different size networks from small size (3×3, 4×4), medium size (5×5, 6×6, 7×7) to large size (8×8, 9×9, 10×10) with one up to eight decision layers respectively.

The addition of Deep Learning Clusters specialised in different functions (Cyber, QoS, and Management) provides a flexible approach similar to how our brain operates; Deep Learning Clusters are able to adapt and be assigned where more routing, computing and memory resources are required. As equivalent to human behaviour, an unstable adaptation of the CPN to QoS changes due to node reward parameter may lead to CPN “anxiety”; and different best routes and QoS metrics from DL and CPN-RNN algorithms due node thresholds adaptation may cause CPN “depression” in the long term.

Conflicts of interest

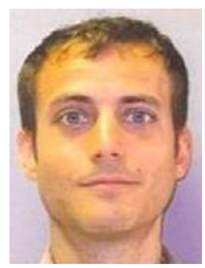
None.

Appendix: CPN with Deep Learning Clusters - Neural Schematic



References

[1] S.B. Danielle, E. Bullmore, Small-world brain networks, Neurosci. Rev. J. Bringing Neurobiol. Neurol. Psychiatry 12 (2007) 512–523.
[2] L.R. Squire, Declarative and non declarative memory: multiple brain systems supporting learning and memory, J. Cognit. Neurosci. 4 (3) (1992) 232–243.
[3] S. Grossberg, The link between brain learning, attention, and consciousness, Conscious. Cognit. 8 (1) (1999) 1–44.
[4] G.N. Ericsson, Cyber security and power system communication, essential parts of a smart grid infrastructure, IEEE Trans. Power Delivr. 25 (3) (2010) 1501–1507.
[5] C.-W. Ten, G. Manimaran, C.-C. Liu, Cybersecurity for critical infrastructures: attack and defense modeling, IEEE Trans. Syst. Man Cybern. 40 (4) (2010) 853–865.
[6] T. Cruz, L. Rosa, J. Proença, L.A. Maglaras, M. Aubigny, L. Lev, J. Jiang, P. Simões, A cybersecurity detection framework for supervisory control and data acquisition systems, IEEE Trans. Ind. Inf. 12 (6) (2016) 2236–2246.
[7] Q. Wang, W. Guo, K. Zhang, A.G. Ororbia II, X. Xing, C.L. Giles, X. Liu, Adversary resistant deep neural networks with an application to malware detection, in: Proceedings of the International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1145–1153.
[8] A. Tuor, S. Kaplan, B. Hutchinson, N. Nichols, S. Robinson, Deep learning for unsupervised insider threat detection in structured cybersecurity data streams, in: Proceedings of the AI for Cyber Security Workshop at Association for the Advancement of Artificial Intelligence, 2017, pp. 4993–4994.
[9] M. Wu, Z. Song, Y.B. Moon, Detecting cyber-physical attacks in Cyber Manufacturing systems with machine learning methods, J. Intel. Manuf. (3) (2017) 1–13.
[10] C. Kim, Cyber–defensive architecture for networked industrial control systems, Int. J. Eng. Appl. Comput. Sci. 2 (1) (2017) 1–9.
[11] Erol Gelenbe. Cognitive Packet Network. Patent US 6804201 B1. (2004)
[12] E. Gelenbe, Z. Xu, E. Seref, Cognitive packet networks, in: Proceedings of the International Conference on Tools with Artificial Intelligence, 1999, pp. 47–54.
[13] E. Gelenbe, R. Lent, Z. Xu, Networks with cognitive packets, in: Proceedings of the IEEE International Symposium on the Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2000, pp. 3–10.
[14] E. Gelenbe, R. Lent, Z. Xu, Measurement and performance of a cognitive packet network, Comput. Netw. 37 (6) (2001) 691–701.
[15] E. Gelenbe, R. Lent, A. Montuori, Z. Xu, Cognitive packet networks: qoS and performance, in: Proceedings of the IEEE International Symposium on the Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2002, pp. 3–9.
[16] Erol Gelenbe, Random neural networks with negative and positive signals and product form solution, Neural Comput. 1 (4) (1989) 502–510.
[17] E. Gelenbe, Stability of the random neural network model, Neural Comput. 2 (2) (1990) 239–247.
[18] E. Gelenbe, Learning with the recurrent random neural network, Int. Fed. Inf. Process. Congr. 1 (1992) 343–349.
[19] E. Gelenbe, F.-J. Wu, Large scale simulation for human evacuation and rescue, Comput. Math. Appl. 64 (12) (2012) 3869–3880.
[20] A. Filippoupolitis, L.A. Hey, G. Loukas, E. Gelenbe, S. Timotheou, Emergency response simulation using wireless sensor networks, Ambient Media Syst. 21 (2008) 1–7.
[21] E. Gelenbe, T. Koçak, Area-based results for mine detection, IEEE Trans. Geosci. Remote Sensing 38 (1) (2000) 12–24.
[22] E. Gelenbe, M. Sungur, C. Cramer, P. Gelenbe, Traffic and video quality with adaptive neural compression, Multimed. Syst. 4 (6) (1996) 357–369.
[23] V. Atalay, E. Gelenbe, N. Yalabik, The random neural network model for texture generation, Int. J. Pattern Recognit. Artif. Intel. 6 (1) (1992) 131–141.
[24] J. Schmidhuber, Deep learning in neural networks: an overview, Neural Netw. 61 (2015) 85–117.
[25] Y. Bengio, A.C. Courville and P. Vincent. Unsupervised Feature Learning and Deep Learning: a Review and New Perspectives. CoRR abs/1206.5538, 1–30 (2012).
[26] S. Jiea, Z. Zhichenga, S. Feia, C. Annia, Progressive framework for deep neural networks: from linear to non-linear, J. China Univ. Posts Telecommun. 23 (6) (2016) 1–7.
[27] Q.V. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, A.Y. Ng, On optimization methods for deep learning, in: Proceedings of the International Conference on Machine Learning, 2011, pp. 265–272.
[28] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, A.Y. Ng, Multimodal deep learning, in: Proceedings of the Multimodal Deep Learning International Conference on Machine Learning, 2011, pp. 689–696.
[29] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2014, pp. 3104–3112.
[30] A.J. Bekker, I. Opher, I. Lapidot, J. Goldberger, Intra-cluster training strategy for Deep Learning with applications to language identification, in: Proceedings of the Machine Learning for Signal Processing, 2016, pp. 1–6.
[31] E. Gelenbe, Y. Yin, Deep learning with random neural networks, in: Proceedings of the International Joint Conference on Neural Networks, 2016, pp. 1633–1638.
[32] Y. Yin and E. Gelenbe. Deep Learning in Multi-Layer Architectures of Dense Nuclei. CoRR abs/1609.07160. 1–6, (2016)
[33] E. Gelenbe, G-Networks: a unifying model for neural nets and queueing networks, in: Proceedings of the Modeling, Analysis, and Simulation On Computer and Telecommunication Systems, 1993, pp. 3–8.
[34] J.-M. Fourneau, E. Gelenbe, R. Suros, G-Networks with multiple class negative and positive customers, in: Proceedings of the Modeling, Analysis, and Simulation On Computer and Telecommunication Systems, 1994, pp. 30–34.
[35] E. Gelenbe, S. Timotheou, Random neural networks with synchronized interactions, Neural Comput. 20 (9) (2008) 2308–2324.
[36] W. Serrano, E. Gelenbe, The deep learning random neural network with a management cluster, in: Proceedings of the International Conference on Intelligent Decision Technologies, 2017, pp. 185–195.
[37] Z. Li and J. Zhang. An Architecture of Network Artificial Intelligence (NAI). Internet Engineering Task Force Draft, Huawei Technologies (2016).
[38] H. Zhang, W. Quan, H.-c. Chao, C. Qiao, Smart identifier network: a collaborative architecture for the future internet, IEEE Netw. 30 (3) (2016) 46–51.
[39] J. Qadir, K.-L.A. Yau, M.A. Imran, Q. Ni, A.V. Vasilakos, IEEE access special section editorial: artificial Intelligence enables networking, IEEE Access 3 (2015) 3079–3082.
[40] W. Quan, Y. Liu, H. Zhang, S. Yu, Enhancing crowd collaborations for software defined vehicular networks, IEEE Commun. Mag. 55 (8) (2017) 80–86.



Will Serrano is a Chartered Engineer and Technology Designer specializing the technical design and design management or large and high-profile infrastructure projects, from airports, rail stations to buildings and energy plants such as CrossRail, Heathrow Gatwick Airports and High Speed 2. He has delivered solutions in Local Area Networks; Wide Area Networks; Wireless LAN; Voice; Cloud and Cybersecurity. Will Serrano holds a Master's Degree in Telecommunication Systems and Networks and a PHD at Imperial College London. He is a Member of the Institution of Engineering and Technology. He is researcher with the Intelligent Systems and Networks Department at Imperial College London, UK.

Erol Gelenbe is a Fellow of IEEE, ACM and IET (UK), and a Professor in the Institute of Theoretical and Applied Computer Science of the Polish Academy of Sciences, and at Imperial College. He has introduced computer and network performance models based on diffusion approximations, and invented the Random Neural Network Model, as well as G-Networks which are analytically solvable queueing models that incorporate control functions such as work removal and load balancing. His other contributions include the concept and prototype for FLEXSIM, an object oriented discrete event simulation approach for flexible manufacturing systems, and other commercially successful projects such as the QNAP tool for the Performance Evaluation of Computer Systems and Networks. His innovative designs include the first voice-packet switch SYCOMORE, the first fibre optics random access network XANTHOS, and the Cognitive Packet Network and its adaptive routing protocol. He also designed and published the first optimal protocol for random access communications, and an optimum checkpointing scheme for databases. He has been awarded several prizes from France, the UK, Hungary and Turkey, including the 2010 IET Oliver Lodge Medal, the 2008 ACM SIGMETRICS Life-Time Achievement Award, and the 1996 Grand Prix France Telecom of the French Academy of Sciences. He was awarded Knight of the Legion of Honour and Officer of the Order of Merit of France, and Grand Officer of the Order of the Star and Commander of Merit of Italy. He is a Fellow of the French National Academy of Engineering, the Royal Academy of Belgium, the Science Academies of Hungary and Poland, and the Science Academy of Turkey. He was awarded Honoris Causa doctorates from the Universities of Liege, Belgium, Roma II, Italy, and Bogazici,Istanbul. He has graduated over 83 PhD students. His recent papers appear in the IEEE Systems Journal, IEEE Access, IEEE Trans. on Selected Areas in Communications, IEEE Trans. on Wireless Networks, Physical Review, and other journals.