Integrated Deep Auto-Encoder and Q-learning-based Deep Learning Scheme to detect anomalies from log data for supporting forensics in Cloud Computing

Savaridassan P (savari.aspire30@gmail.com)
SRM Institute of Science and Technology
Maragatham G.
SRM Institute of Science and Technology

Research Article

Keywords: Bio-inspired algorithm, Q-learning, Deep Auto-Encoder, Anomaly detection

Posted Date: March 29th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-323410/v1

Version of Record: A version of this preprint was published at Wireless Personal Communications on August 8th, 2021. See the published version at https://doi.org/10.1007/s11277-021-08785-6.

Integrated Deep Auto-Encoder and Q-learning-based Deep Learning Scheme to detect anomalies from log data for supporting forensics in Cloud Computing

Savaridassan P[1*], savaridp@srmist.edu.in[1*], Assistant Professor[1*]
Dr. Maragatham G[2], maragatg@srmist.edu.in[2], Associate Professor[2]
[1, 2] Department of Information Technology, School of Computing, College of Engineering and Technology,
Faculty of Engineering and Technology, SRM Institute of Science and Technology,
SRM Nagar, Kattankulathur, Chengalpattu – 603203, Chennai, Tamil Nadu, India

Abstract: The cloud computing environment when deployed correctly is responsible for delivering scalability, cost efficiency, reliability, security and interoperability to the end users. Log analysis is considered to be an indispensable component of security regulations and framework, since these computer-generated records help the organizations, businesses and networks to respond to different kinds of risks that are possible to cloud environment in a reactive and proactive manner. In this paper, an Integrated Deep Auto-Encoder and Q-learning-based Deep Learning (IDEA-QLDL) Scheme is proposed for attaining maximum prediction accuracy during the process of exploring log data and classifying them into genuine and anomalous. It initiates the process of acceptance or denial based on the continuous investigation of behavioral patterns that are highly applicable for classification. The results of the proposed IDEA-QLDL Scheme confirmed its predominance in improving the classification accuracy, precision, recall and detection time compared to the benchmarked schemes considered for investigation.

Keywords: Bio-inspired algorithm, Q-learning, Deep Auto-Encoder, Anomaly detection

1. Introduction

From the past decade, cloud computing is considered as one of the most potential technologies in the modern era due to its significance in satisfying the user requirements with increased on-demand services and improved computational power [1]. As per the reports of Gartner, cloud computing is determined to be growing at a rapid rate of 40% and still has the possibility of continuing its progressive growth at a rate greater than 28% per year [2]. IDS schemes proposed for cloud computing with the objective of detecting anomaly from system logs based on the heterogeneous traffic flow data generated through different types of application is determined to be still in the stage of infancy [3].From the recent past, diversified categories of anomaly detection schemes integrated with IDSs were contributed in the literature for combating security challenges in cloud computing [4]. For example, Artificial Neural Network and Fuzzy C-Means clustering algorithm-based IDS were proposed for detecting attacks at the layer of hypervisor in the cloud computing scenario. Likewise, a multi-order Markov chain-based anomaly detection scheme was proposed and explored using DARPA dataset, Further, Multivariate Correlation Analysis (MCA)-based anomaly detection scheme was proposed for detecting unknown and known Dos attacks in clouds. In addition, a one-class support vector machine algorithm-based anomaly detection scheme was proposed for online cloud in order to detect

different DOS attacks and different kind of malwares in cloud computing infrastructures. However, most of these techniques elevated computation complexity and suffered from high false alarm rates in spite of their competencies in detecting anomalies from system logs in cloud scenario. Moreover, majority of the existing works in the literature were not potent when they especially used for network anomaly detection under streaming data that necessitates the analysis in real time [5]. Then, deep learning conquered the attention of huge numbers of researcher for using them into the process of detecting anomaly in cloud computing. This deep learning is one of the widely utilized machine learning technique that plays an anchor role in identifying the most relevant features from the complex datasets through back propagation process. From the inception of deep learning, diversified architecture variants such as Convolution Neural network, Recurrent Neural Network, Deep Belief Network networks, Deep Neural network and Autoencoders were contributed in the literature. Among the deep learning architectures, Autoencoders are considered to be highly predominant due its capability in efficient data classification. This merit of data classification is mainly achieved through the training attained with minimum preprocessing requirements, which makes them highly suitable for detecting anomaly from logs of cloud computing [6].

Motivation of the work

The above discussion evidently portrayed that different proposals using Multivariate Correlation Analysis (MCA), multi-order Markov chain and, Artificial Neural Network and Fuzzy C-Means clustering algorithm were proposed for detecting anomalous characteristics from network traffic. But, the contributed techniques in the literature were determined to be inefficient due to its high false positive alarms with limitation of minimized accuracy. Further, the available schemes may not be suitable in resolving the challenges introduced by the existence of virtualized scenario and diversified kinds of application workloads as the cloud environments possess heterogeneous characteristics [7]. This IDEA-QLDL model is developed for log-based anomaly detection that especially handles heterogeneous data in cloud computing networking scenario. It was proposed for exploring two important issues such as, i) selection of relevant feature set from the repository of traffic stream and ii) classification of data traffic streams into benign and anomalous classes for attack detection.

The major contributions of this proposed IDEA-QLDL scheme is presented as follows.

i) It is proposed as an efficient integrated model for efficient anomaly detection in cloud setups using the benefits of SFHOA and Q-Learning improved deep Autoencoders architecture.
ii) It utilized SFHOA for multi-objective feature extraction and Q-Learning improved deep Autoencoders architecture for detecting anomalous behavior from logs.
iii) It utilized Deep Auto-Encoder and Q-learning-based Deep Learning for attaining continuous learning of behavior patterns of massive logs for achieving better predictions with better accuracy.

iv) The quantitative and qualitative investigations of the proposed scheme and the classical state-of-art log-based anomalous detection approaches are conducted based on baseline and synthetic datasets.

2. Related work

In this section, the review of the existing works contributed towards log-based anomaly detection is presented with the merits and limitations.

A Long Short-Term Memory (LSTM)-based deep network model named DeepLog was proposed by Du et al. [8] for modeling the system into a sequence of natural language in order to detect anomalous behavior. This DeepLog was proposed for automatic learning of log patterns from the normal operation in order to identify anomalies when the patterns of logs deviate from the trained model. It was developed with the merits of updating the DeepLog model incrementally in a dynamic manner such that it can adapt itself for learning log patterns of network for a considered amount of time. It was designed to derive system log through which the workflows may be constructed depending on the detection of anomaly for the purpose of enforcing root cause analysis in an efficient manner. The extensive experimental analysis of DeepLog conducted using huge amount of log data confirmed its significance in terms of accuracy, precision, recall and F-Measure over the existing log-based anomaly detection strategies. However, this DeepLog strategy is considered to still have the probability of improvement in accuracy and F-Measure. A stacked Long Short-Term Memory (Stacked-LSTM)-based deep network model named nLSALogwas proposed by Yang et al. [9] for detecting anomaly by converting the system logs into a sequence of natural language. This nLSALog was proposed with the mechanism of self-attention such that it can potentially extract the hidden patterns associated with the sequence of log template with the view to determine their dependencies. It was designed for both offline and online anomaly detection as it prevents the use of any input sample inversion approach. The simulation results of nLSALog proved its predominance in lower time cost and overall accuracy on par with the existing anomaly detecting frameworks of the literature. However, the computational complexity of this nLSALog was comparatively lower depending on the number of logs generated in cloud computing scenario.

A log-based anomaly detection scheme using Log-Normal Distributed Stochastic Neighbor Embedding (LN-SNE) was proposed by Ghafoori et al. [10] with unsupervised dimensionality reduction for achieving better detection accuracy. This LN-SNE approach used Restricted Boltzmann Machines for generating parametric embedding that aided in handling the imbalance visualized on the dataset. It further included a heavy-tail distribution with data projected into a lower space of dimensionality for attaining better discrimination between genuine and anomaly data. This utilization of heavy-tail distribution also strengthened and preserved the lower dimensional space during anomaly detection process. The experimental results of this LN-SNE approach was determined to be less sensitive in the latent space dimension with improved accuracy level. The empirical results of LN-SNE approach was also determined to non-linearly

scale up with different data size and the number of dimensions. However, the time cost involved in detection has still the possibility of improvement independent to the size of systems logs used for detection. An integrated k-NN classification and K-prototype clustering scheme was proposed by Liu et al. [11] for facilitating anomaly detection over massive logs. This anomaly detection scheme was proposed for exploring the system logs characteristics by extracting ten features based on the information of the session that effectively inherits user behaviors. It used the merits of K-prototype clustering algorithm by extracting features that aided in partitioning dataset into different number of clusters, such that highly correlated clusters could be determined for exploring them as anomaly candidates. It also utilized the degree of global and local distance-based features for subsequent application of k-NN classifier into the determination of accurate detection results. The experiments confirmed lower computational complexity and high detection accuracy independent to the benchmarked and synthesized datasets considered for analysis. A LogEvent2vec-based anomaly detection scheme was proposed by Wang et al. [12] with reduced number of transformations and computation cost involved in exploring log data. This detect scheme considered the input of word2vec as log event and helped in determining the relevance between vectorized log events and normal log events in a more direct manner. It specifically used LogEvent2vec for anomaly detection as it incorporates the capability of being integrated with any coordinate transformation methods. It further used the models of neural networks, naive Bayes and random forests for training and detecting anomalies by converting log event vector to log sequence vector. The experimentsof theLogEvent2vec-based anomaly detection scheme conducted using a real time public log BlueGene/L (BGL) dataset was determined to improve accuracy and reduce time incurred in computation to the considerable level. However, this predominance in reduced computation time and improved accuracy was not realized with large system logs.

Furthermore, an integrated Autoencoders and isolation forest-based log message anomaly detection scheme was proposed by Farzad et al. [13] for attaining superior accuracy with least computation complexity. This anomaly detection approach utilized isolation forest for predicting positive samples and autoencoders for training and extracting significant features. The experiments of this anomaly detection scheme conducted using the datasets of Thunderbird log, Openstack and BGL confirmed that the negative samples count prediction is comparatively lower than the benchmarked deep learning approaches. The results also proved that the computational complexity and time incurred in feature extraction is considerably minimized independent to the size of the dataset considered for investigation. A security analytical framework named PredictDeep was proposed by Elsayed and Zulkernine [14] for facilitating better prediction and anomaly detection from log data. This PredictDeep framework leveraged the benefits of graph analytics and deep learning for collecting log data and exploring them for efficient discrimination of normal logs from malicious logs. This framework initially collected log data and modeled them into a graph for capturing analytical activities and correlation that aids in better differentiation. This model utilized context information and graph structure for extracting potential features which aids in boosting prediction and anomaly classification. It also leveraged the benefits of graph embedding for highlighting the nodes and its associations from the graph model for the objective

of learning and predicting anomalies from feature vectors. The experiments of this PredictDeep framework confirmed better accuracy and reduced exploration time for log data investigation. However,PredictDeep framework and the isolation forest-based log message anomaly detection schemes fails in attaining maximized F-Score and detection rate with massive logs data. Wang et al [15] proposed another log-based anomaly detection strategy with improved k-nearest neighbor for reducing detection time with enhanced overall accuracy. This anomaly detection approach utilized the classical mean shift clustering algorithm for improving the efficiency of KNN algorithm towards ideal classification. The utilization of mean shift clustering algorithm was identified to facilitate better efficiency in selecting training set from massive logs. It also reduced the negative impact of unbalanced log samples distribution by assigning adaptive weights to samples with different distances. This log-based anomaly detection strategy was visualized to attain better accuracy, recall rate and F-measure compared to the conventional anomaly detection approaches.

3. Proposed Integrated Deep Auto-Encoder and Q-learning-based Deep Learning Scheme

An Integrated Deep Auto-Encoder and Q-learning-based Deep Learning (IDEA-QL-DL) Schemeutilized for log-based anomaly detection is presented with its merits. This proposed scheme utilizes the potential of Q-learning improved deep auto-encoders with the optimization of features achieved through SailFish Hunting-based Optimization Algorithm (SFHOA) for detecting anomalies from log-based data.

3.1 SailFish Hunting-based Optimization Algorithm (SFHOA)

This SHOA algorithm is considered to be directly applicable for the problems of optimization without any requirements of structural modifications. SFHOA is proposed based on the inspiration derived from the group hunting behavior of sailfish.

i) Population Initialization

SHFOA is a population-based metaheuristic algorithm with sailfishes considered as the candidate solutions (search agents used for feature optimization) and the variables of the problem as the sailfish position (the parameters considered for optimization) from the search space. Thus, the population is randomly generated over the solution space in order to facilitate the process of searching in the one, two or hyper-dimensional space with their variable position vectors. In the k^{th} searching bout, the member 'i' in a d-dimensional search space possess a current position $SF_{Pos(i,d)} \in R$ with $1 \le i \le n$. Each individual sailfish (search agent) evaluates the fitness value of all candidate solutions based on Equation (1)

$$SF_{Fit\_Pos(i,d)} = \begin{bmatrix} f(SF_{Pos(1,1)}, SF_{Pos1,2)}, ...SF_{Pos(1,d)}) \\ f(SF_{Pos(2,1)}, SF_{Pos(2,2)}, ...SF_{Pos(2,d)}) \\ .... \\ f(SF_{Pos(n,1)}, SF_{Pos(n,2)}, ..., SF_{Pos(n,d)}) \end{bmatrix} = \begin{bmatrix} F(SF_{Pos(1)}) \\ F(SF_{Pos(2)}) \\ ..... \\ F(SF_{Pos(n)}) \end{bmatrix} \quad (1)$$

Where, ' f ' is the fitness function used for determining the value of i^{th} sailfish (primary feature) in the j^{th} dimension (variables considered for evaluating primary features).

In this scenario, the secondary features that need to be optimized (considered as the school of sardines) is another potential community that also swims in the same search space of sailfish. Then, each individual sardine (search agent used for secondary feature optimization) evaluates the fitness value of all secondary candidate solutions based on Equation (2)

$$SA_{Fit\_Pos(i,d)} = \begin{bmatrix} f(SA_{Pos(1,1)}, SF_{Pos1,2)}, ...SF_{Pos(1,d)}) \\ f(SA_{Pos(2,1)}, SF_{Pos(2,2)}, ...SF_{Pos(2,d)}) \\ .... \\ f(SA_{Pos(m,1)}, SF_{Pos(m,2)}, ..., SF_{Pos(m,d)}) \end{bmatrix} = \begin{bmatrix} F(SA_{Pos(1)}) \\ F(SA_{Pos(2)}) \\ ..... \\ F(SA_{Pos(m)}) \end{bmatrix} \quad (2)$$

Where, ' f ' is the fitness function used for determining the value of i^{th} sardine (secondary feature) in the j^{th} dimension (variables considered for evaluating secondary features).

At this juncture, it is identified that both the sailfish (primary features) and sardines (secondary features) are associated with one another in the process of determining the optimal solution in the optimization process. However, the sailfish (search agents for exploring primary features) need to be supported by sardines (search agents for exploring secondary features) for finding the core features that aids in determining better classification results with maximized accuracy. Moreover, the sailfish (search agents used for exploring primary features) can determine potential features that dominate over the sardines (search agents used for exploring secondary features) when the better solution is determined by sailfish search agents in search space.

ii) Employment of elitism

The employment of elitist selection is mainly responsible for preventing the loss of good solutions (predominant optimal features) during the process of search agent updation. This elitism impacts the acceleration and maneuverability of the sardines during the process of optimization. The best solution (elite) of sailfish ($OS^{i}_{Elite\_SF}$) and recessive solution of sardines ($OS^{i}_{Best\_SA}$) are considered to possess remarkable value in each iteration.

iii) Strategy of attack alternation

It is visualized that sailfish search agents facilitate the phase of exploration by including large search space in order to identify the predominant solutions (vital features) that can be still improved based on exploitation process. In this SFHOA algorithm, the optimal solution $OS^{i}_{New\_SF}$ of sailfish at the r^{th} iteration is updated based on Equation (3)

$$OS^{i}_{New\_SF} = OS^{i}_{Elite\_SF} - \alpha_i (rand(0,1) \times (\frac{OS^{i}_{Elite\_SF} - OS^{i}_{Bset\_SA}}{2}) - OS^{i}_{Old\_SF}) \quad (3)$$

Where, α_i is the r^{th} iteration coefficient computed based on Equation (4)

$$\alpha_i = 2 \times rand(0,1) \times (D_p - IS_p) \qquad (4)$$

Where, $rand(0,1)$ is the random number that lies between 0 and 1. 'D_p' represent the density of prey which portrays the prey count at each and every individual iteration. IS_p is the significant parameters considered in updating the solution of sailfish around the neighborhood solutions determined by sardines search agent. This adaptive factor of D_p is determined based on Equation (5)

$$D_p = 1 - (\frac{SF_N}{SF_N + SA_N}) \quad (5)$$

Where, SF_N and SA_N are the number of sailfish agents and number of sardine search agents used for feature optimization processes. In this strategy of attack alternation, sailfish search agents can achieve better exploration based on the coefficient α_i defined in Equation (4). This coefficient α_i is considered to vary between -1 and +1, but completely depends on the sardines prey count and $rand(0,1)$.

iv) Prey hunting and catching strategy.

In this context, the new position of the sardine search agent ($OS^{i}_{New\_SA}$) at the i^{th} iteration is updated based on Equation (6)

$$OS^{i}_{New\_SA} = r_d \times (OS^{i}_{Elite\_SF} - OS^{i}_{Old\_SA} + AP_{SA}) \qquad (6)$$

Where, $OS^{i}_{Elite\_SF}$ and $OS^{i}_{Old\_SA}$ corresponds to the position of elite sailfish search agent and old position of the sardine search agent. r_d is the random value that ranges between 0 and 1. AP_{SA} Is the attack power possessed by the sardine search agent is calculated based on Equation (7)

$$AP_{SA} = C_f \times (1 - (2 \times I_{tr} \times \delta)) \quad (7)$$

This attack power possessed by the sardine search agent (AP_{SA}) depends on the coefficients C_f and δ, which is decreased linearly from C_f to 0. Further, r_d is the random value used by the search agents to regulate the exploration rate of solutions that exists in their neighborhood. Equation (7) permits the sardine search agent to escape from the elite sailfish for the objective of exploiting information with respect to their associated neighbors in order to facilitate a reliable local search (exploitation). Hence, this SFHOA algorithm plays a vital role in sustaining the balance between the exploitation and exploration in the search space. Moreover, the

sardine search agents using the parameter AP_{SA} to update the position (ψ) and the number of associated variables (λ) based on Equation (8) and (9)

$$\psi = SA_N \times AP_{SA} \quad (8)$$

$$\lambda = N_{Var(i)} \times AP_{SA} \quad (9)$$

Where, N is the number of variables in the i^{th} iteration with SA_N as the sardine search agent count in each individual cycle of the algorithm. When the value of AP_{SA} is less than 0.5, then ψ sardines with λ sardine variables is considered to be updated. On the other hand, the complete set of sardines position is updated when the value of AP_{SA} is greater than 0.5. Thus, the parameters of AP_{SA} and r_d plays an indispensable role in portraying more random characteristics during the entire optimization process. These parameters are also responsible for attaining stagnation in local optima during the complete set of iterations. In this phase, the optimal set of features is determined when the solution estimated by the sardine search agent is fitter than the solution determined by the corresponding sailfish search agent. At this juncture, the selfish agents' position is replaced with the hunted sardine search agents' current position in order to increase its potentiality in determining potential features that aids in better classification accuracy during log-based anomaly detection in cloud scenario.

3.2 Q-learning-based Deep Auto-Encoder Model used for anomaly classification

This hybrid log-based network-wide anomaly detection model is accomplished in two phases and their detailed operation is presented as follows.

3.2.1 SFHOA algorithm-based feature selection and optimization

In this phase, the performance of the classifier completely depends on the potential number of features such as source port number, source IP address, destination port number and destination IP address., etc. This problem of anomaly detection comprises of determining the most relevant and significance in order to maximize the performance of the classifier. If $D = \{x_1, x_2, ... x_n\}$ and $F = \{f_1, f_2, ... f_n\}$ be the well balanced dataset with atmost number of 'k' objects and the set of features with 'l' number of features, respectively, then the process of feature selection and optimization is identified as a mapping represented as $M(D, F, D_{Attr}) \rightarrow FS_{A\lg}$. Where, D_{Attr} and $FS_{A\lg}$ represents the class labels of decision attributes and the utilized feature selection algorithm.

Fitness function of Sailfish:

The process of feature selection during the task of network anomaly detection is considered to suffer from two conflicting objectives such as, i) Reducing the classification error rate and ii) number of features minimization. The utilized SFHOA algorithm targets in computing a feature subset that achieves lower rate of classification error (ξ). In this paper, accuracy is used

for evaluating the classification error rate. Moreover, the fitness function of this problem concentrates on minimizing ξ presented in Equation (10).

$$\xi = \frac{F_P + F_N}{T_P + T_N + F_P + F_N} \quad (10)$$

Where, T_P, T_N. F_P and F_N represents the true positive, true negative, false positive and false negative. This primitive function is modeled as a multi-objective fitness function and presented in Equation (11) with the first objective function ($FS_{(1)}$) targeting on classification error rate minimization and the second objective function ($FS_{(2)}$) concentrating on the reduced number of features.

$$FS_{AIg} = \begin{Bmatrix} CE(\xi) & FS_{(1)} \\ \varpi \times \# \frac{FS_{Avg}}{F} + (1-\omega) \times \frac{\xi_{CER}^{FS_{Avg}}}{\xi_{AVC}^{F}} & FS_{(2)} \end{Bmatrix} (11)$$

ω is any value of constant that lies between 0 and 1. In the implementation, the value of ω is set to 0.5. Where, $\xi_{CER}^{FS_{Avg}}$ and ξ_{AVC}^{F} refers toclassification error rate used for feature set selection and rate of classification error incurred under the use of the complete set of available features.

3.2.2 DAEQN-based anomaly detection

In this IDEA-QL-DL Scheme, the deep auto-encoder consists of more than three hidden layers, in which each hidden layer inherits a finite number of neurons equal to the number of features considered as input. The input parameters of the Deep Auto-Encoder based Q learning Network (DAEQN) are determined based on cross-validation method that explores different dimensions of feature combinations for achieving evaluation that aids in the mitigation of over-fitting risk. This DAEQN network exploits a deep auto-encoder that provides a non symmetrical multiple number of hidden layers for reducing computational overheads and time during the correctness retention of learning structure. It was developed with the significance of acquiring feasible feature patterns with learning characteristics in order to prevent manual interactions. In specific, the two deep auto-encoders incorporated in the proposed IDEA-QL-DL Scheme has different objectives as presented in Figure 1. The first objective focuses on Q-learning-based training process and the second objective concentrateon Q-learning updates that facilitate accurate predictions by minimizing the problems of constantly changing models. In this proposed IDEA-QL-DL Scheme, periodical updating of the current state can be attained through Q-Learning model such that the steps of training. This DAEQN model included into IDEA-QL-DL approach possesses an action-value function in order to estimate the decision based on the integration of state ' S ' and its associated actions ' A ' representing $Q : S \times A \rightarrow R$. In this context the value of is updated based on Equation (12)

$$Q(S_i, A) \leftarrow Q(S_i, A_i) + L_r(r_{wd(t+1)} + D_f \max_{L_r} Q(S_{t+1}, A) - Q(S_t, A_t)) (12)$$

Where, L_r and r_{wd} is the learning rate and reward used for achieving potential pre-training process with the auto-encoder classification architecture.

$$L_{fn}(\theta) = D_{EDM}(S_t, A_t, r_{wd(t+1)}, S_{(t+1)}) \quad (13)$$

Where ' D_{EDM} ' is the experience replay memory used in DAEQN model.

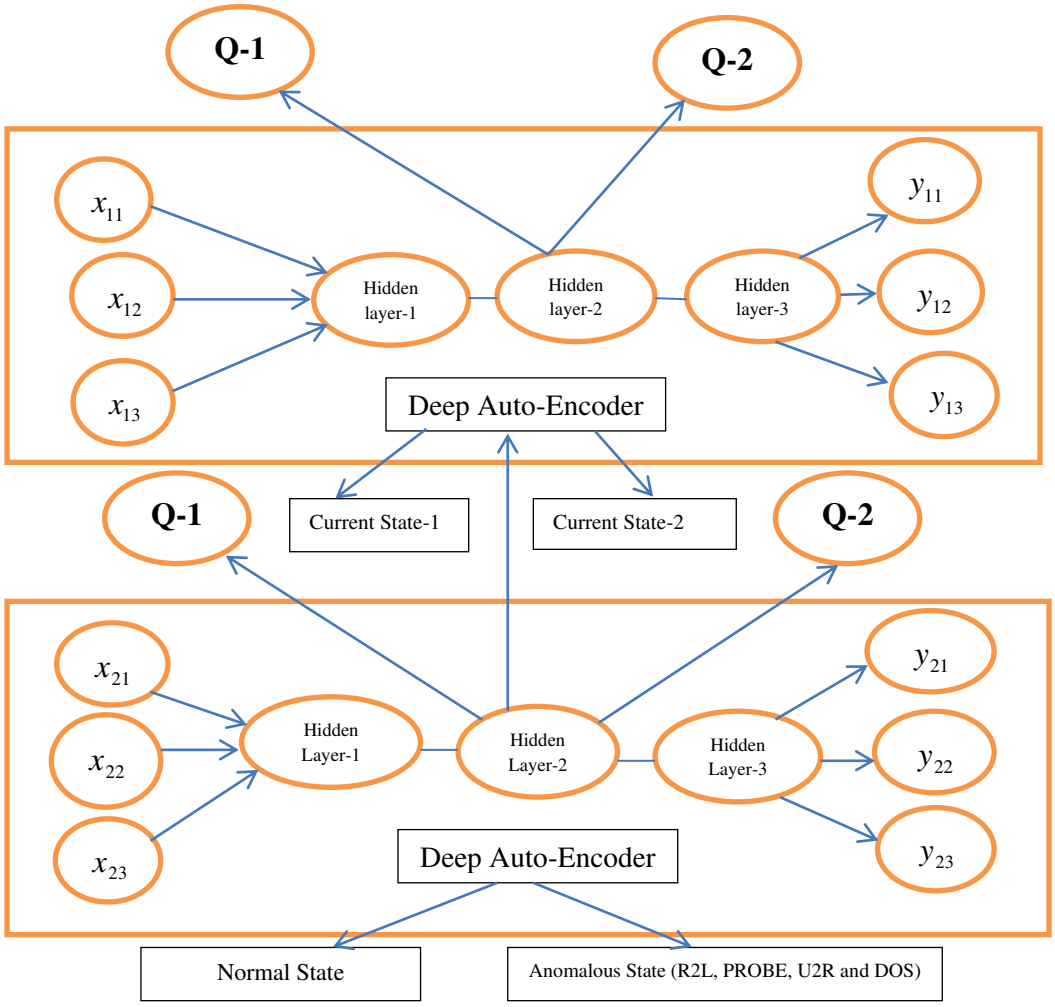

Figure 1: Architecture of DAEQN used for anomaly-based detection

4. Simulation Results and Discussion

In this section, the performance of the proposed IDEA-QL-DL Scheme and the benchmarked PredictDeep, kNN-LAD and IkNN-LAD approaches are evaluated for determining their predominance on the objective of log-based anomaly identification. The experiments of the proposed IDEA-QL-DL Scheme and the baseline approaches are implemented using MATLAB R2016a installed over the system with the configuration of i3-6100U CPU @ 2.30 GHz with 4 GB of RAM. The performance metrics used for evaluating the proposed IDEA-QL-DL Scheme are accuracy, precision, F-Score, detection rate and false positive rate. The mathematical definition of the aforementioned parameters is presented as follows.

$$Accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \qquad (14)$$

$$\Pr ecision = \frac{T_P}{T_P + F_P} \quad (15)$$

$$Detection\_Rate(\Re call) = \frac{T_P}{T_P + F_N} \quad (16)$$

$$False\_Positive\_Rate = \frac{F_P}{F_P + T_N} \qquad (17)$$

$$F - Score = 2 \times \frac{\Pr ecision}{\Pr ecision + \Re call} \qquad (18)$$

Where T_P, T_N. F_P and F_N represents the true positive, true negative, false positive and false negative.

Datasets used for evaluation

The datasets of KDD'99, DARPA and synthetic datasets used for evaluating the proposed IDEA-QL-DL Scheme and the benchmarked approaches are explained as follows. KDD'99 benchmark dataset refers to one of the predominant dataset derived from the UCI machine learning repository. This KDD'99 benchmark dataset consists of approximately 5 million records with a total of 41 features. The traffic visualized in the KDD'99 benchmark dataset can be classified into the classes of R2L, PROBE, U2R and DOS attacks. The DARPA benchmark dataset is derived from Defense Advanced Research Projects Agency. This DARPA dataset consists of totally 58 features and it is considered to be widely considered for evaluating the potential of the proposed approaches contributed towards anomaly detection. The synthetic network traffic schemes are generated by establishing a simulation environment that aids in achieving better comprehensive evaluations of the proposed IDEA-QL-DL Scheme. For the

generation of synthetic datasets, two machines were established with one typical Windows PC and the other machine as a dummy server. The first machine was used for executing different categories of malicious files for the purpose of anomalous traffic generation. On the other hand, INetSim 2 was utilized by the second machine for imitating the operation of Internet. The core benefits of INetSim 2 concentrate on the generation of data such as FTP, DNS, SMTP, and HTTP associated with Internet services. The communication between the server and the first machine consists of benign and anomalous traffic and thus, they can be employed for evaluating the performance of the proposed IDEA-QL-DL model.

The experiments of the proposed IDEA-QL-DLScheme and the benchmarked PredictDeep, kNN-LAD and IkNN-LAD approaches are conducted in four folds based on the evaluation metrics of accuracy, F-Measure, detection time with respect to normal and malicious (R2L, PROBE, U2R and DOS) attacks. Initially, Figure 2 and 3 presents the accuracy of the proposed IDEA-QL-DL Scheme with benchmark KDD'99 dataset is considered to be improved by 2.32%, 3.94% and 4.92%, compared to the baseline PredictDeep, kNN-LAD and IkNN-LAD approaches. Likewise, the accuracy of the proposed IDEA-QL-DL Scheme with benchmark DARPA dataset is considered to be improved by 1.94%, 2.98% and 3.52%, compared to the baseline approaches.

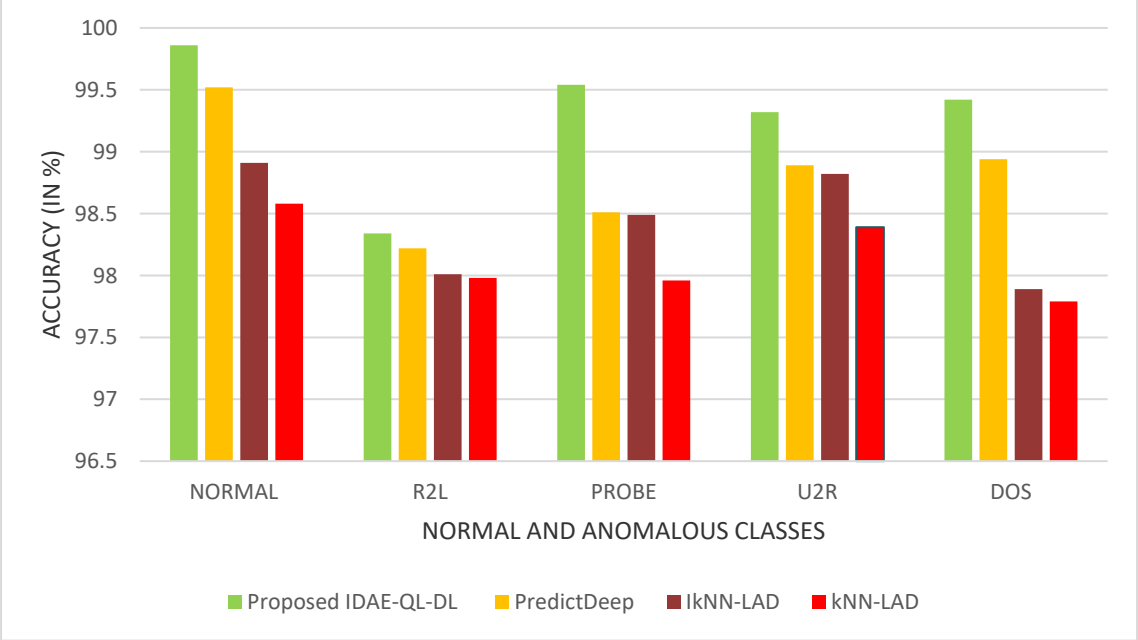

Figure 2:IDEA-QL-DLScheme-Accuracy in anomaly detection with KDD'99 dataset

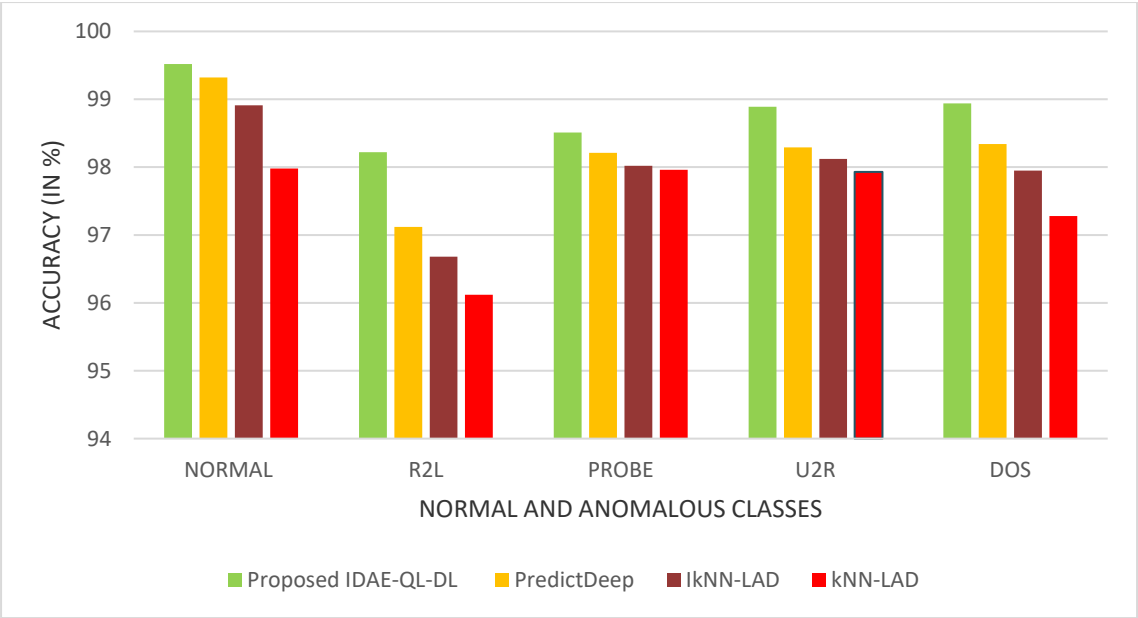

Figure 3:IDEA-QL-DLScheme-Accuracy in anomaly detection with DARPA'98 dataset

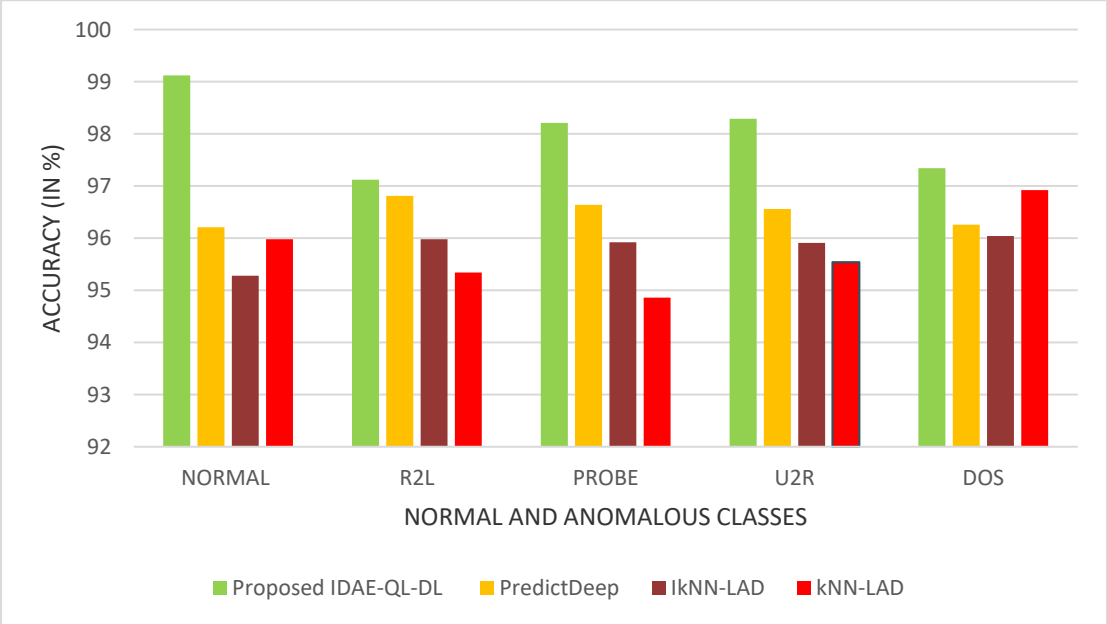

Figure 4:IDEA-QL-DLScheme-Accuracy in anomaly detection with synthesized dataset

Figure 4 demonstrates the accuracy of the proposed IDEA-QL-DL Scheme with synthetic dataset is considered to be improved by 7.36%, 8.14% and 9.42%, compared to the baseline approaches.

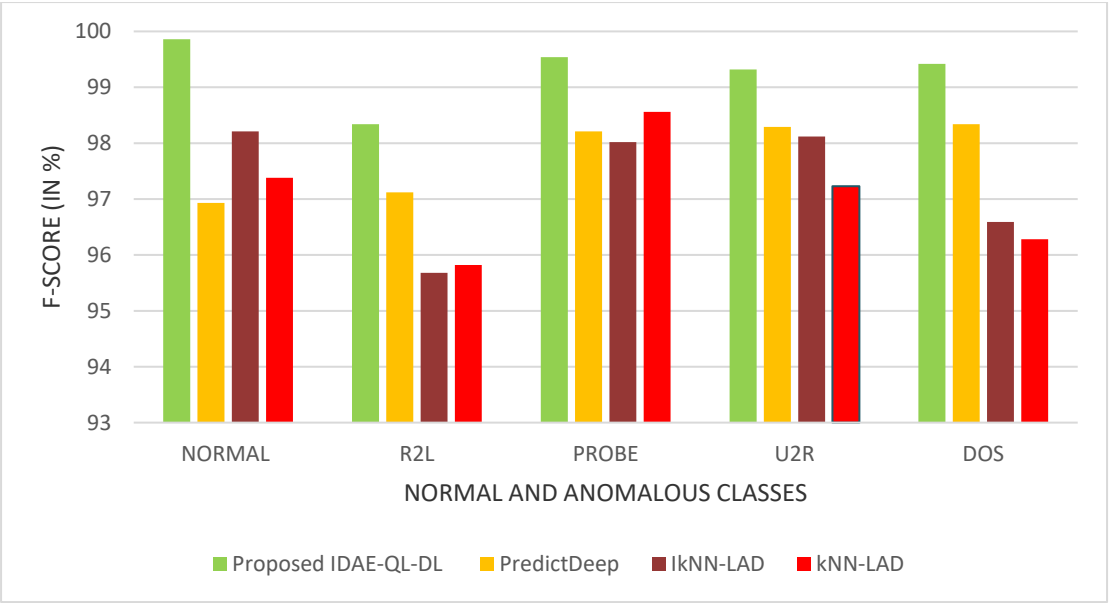

Figure 5:IDEA-QL-DLScheme-F-Score in anomaly detection with KDD'99 dataset

In the second part of analysis, Figure 5 and 6 depicts the F-Score of the proposed IDEA-QL-DL Scheme with benchmark KDD'99dataset is considered to be improved by 7.94%, 9.12% and 10.94%, compared to the baseline PredictDeep, kNN-LAD and IkNN-LAD approaches. Likewise, the F-score of the proposed IDEA-QL-DL Scheme with benchmark DARPA dataset is also identified to be enhanced by 6.84%, 8.42% and 10.68%, compared to the baseline approaches.

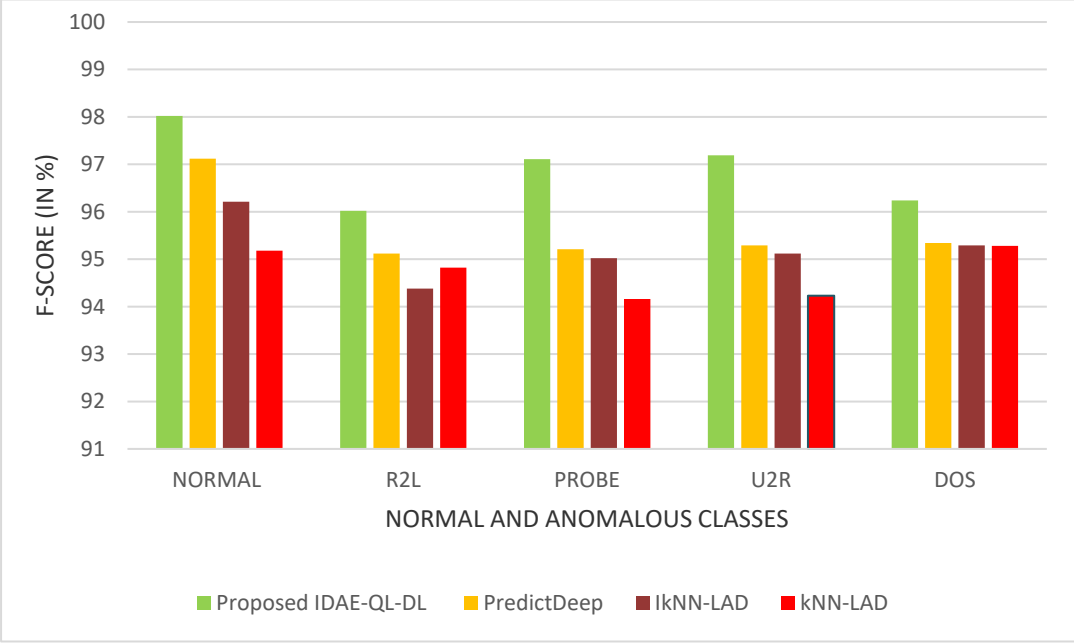

Figure 6:IDEA-QL-DLScheme-F-Score in anomaly detection with DARPA'98 dataset

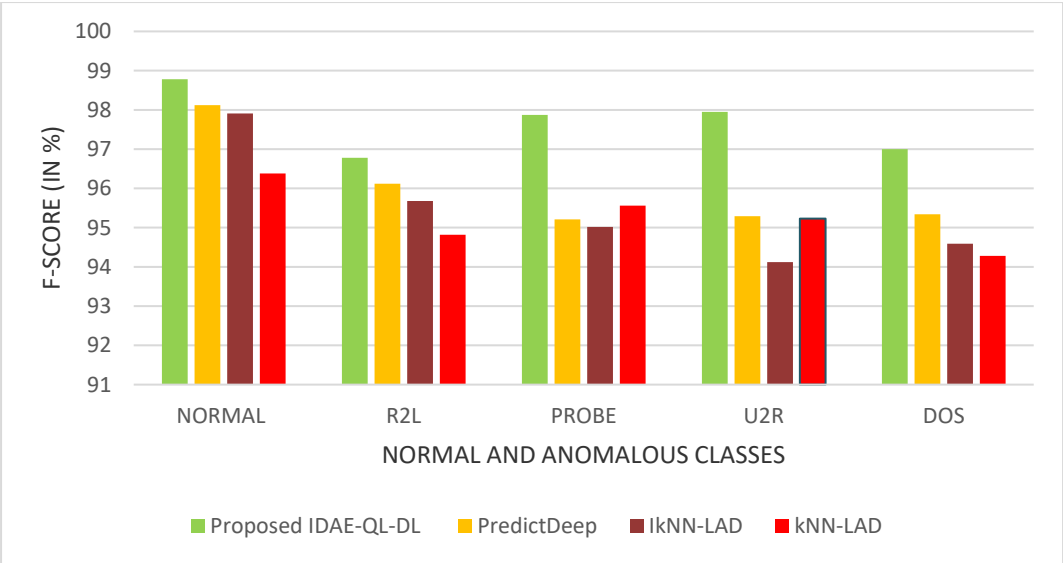

Figure 7:IDEA-QL-DLScheme-F-Score in anomaly detection with synthesized dataset

Figure 7 exemplars the F-Score of the proposed IDEA-QL-DL Scheme with synthetic dataset is considered to be improved by 8.28%, 9.74% and 11.26%, compared to the baseline approaches.

In the final part of the investigation, the proposed IDEA-QL-DL Scheme with KDD'99 datasets is identified to be enhanced by 9.34%, 11.28% and 13.19%, better than the baseline approaches used for investigation. The detection rate of the proposed IDEA-QL-DL Scheme with DARPA dataset is also realized to be enhanced by 9.54%, 11.24% and 12.82%, superior to the benchmarked approaches.

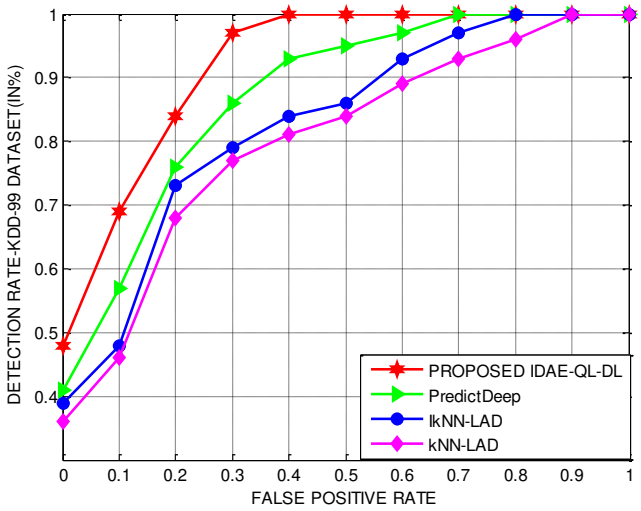

Figure 8:IDEA-QL-DLScheme-Anomaly Detection rate Vs. False positive rate with KDD'99 dataset

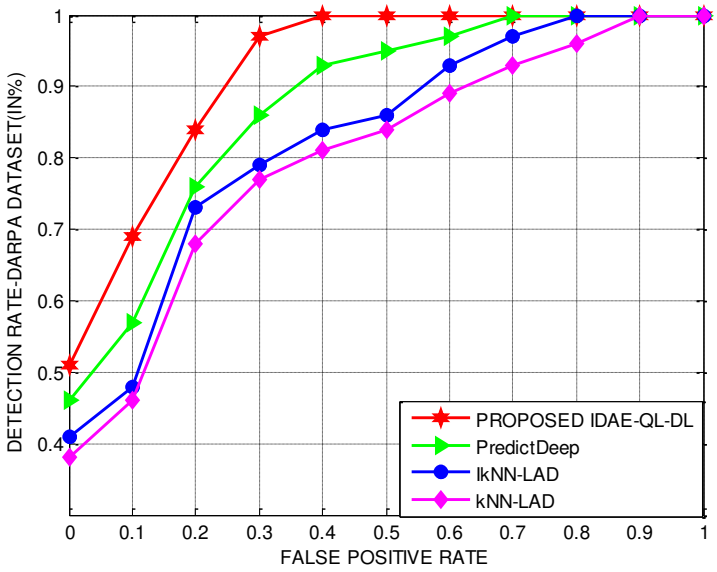

Figure 9:IDEA-QL-DLScheme-Anomaly Detection rate vs False positive rate with DARPA'98 dataset

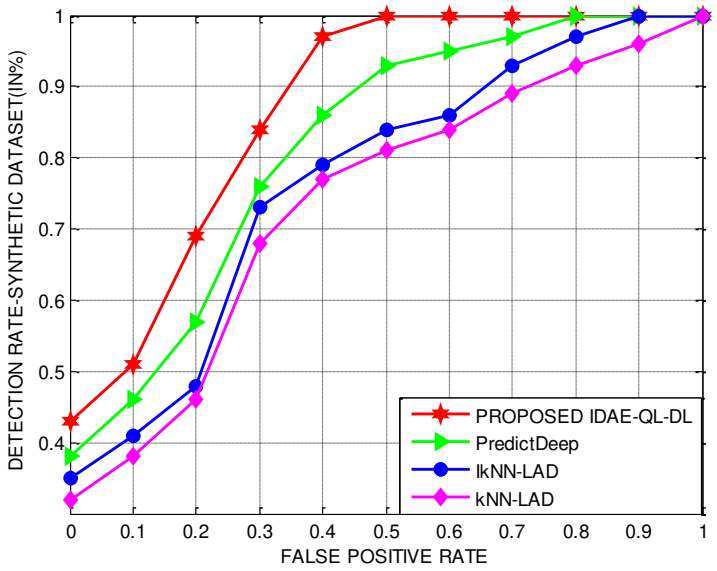

Figure 10:IDEA-QL-DLScheme-Anomaly Detection rate vs False positive rate with synthesized dataset

Figure 10 depicts the detection rate of the IDEA-QL-DL Scheme with KDD'99 datasets is identified to be enhanced by 9.34%, 11.28% and 13.19%, better than the baseline approaches used for investigation.

Table 1 highlights the time incurred for anomaly detection is comparatively lower than the reviewed works of the literature, since the utilization of SFHOA for optimization has prevented insignificant features from being considered during the event of classifying normal classes and malicious classes.

Table 1:Time for anomaly detection incurred by IDEA-QL-DL and the reviewed approaches

Anomaly detection Scheme	Time incurred in anomaly detection (seconds)	
	KDD'99 dataset	**DARPA dataset**
Du et al. [8]	7.98	7.74
Yang et al. [9]	7.92	7.89
Ghafoori et al. [10]	8.16	8.12
Liu et al. [11]	7.94	7.82
Wang et al. [12]	8.48	8.21
Farzad et al. [13]	7.64	7.98
Elsayed and Zulkernine [14]	8.12	7.48
Wang et al. [15]	7.82	7.54
Proposed IDEA-QL-DL	**6.56**	**6.39**

5. Conclusions

In this paper, IDEA-QLDL scheme was presented with merits of SFHOA and Q-Learning-improved deep auto-encoder architecture for detecting anomalies from log data with maximized classification accuracy and reduced computation complexity. In specific, the proposed IDEA-QLDL scheme inherited SFHOA for improving exploitation, exploration and initial population generation capabilities. The Q-Learning-improved deep auto-encoder architecture further facilitated better decisions in classifying logs into normal and anomalous by continuous learning of behavior patterns for achieving better predictions with better accuracy. It was developed with the potential that transformed unstructured log data for training significant features that plays an anchor role in classification processes. The results confirmed that the accuracy. precision and recall of the proposed IDEA-QL-DL Scheme, on an average is enhanced by 6.32%, 7.58% and 8.26%, respectively compared to the baseline approaches used for investigation. The results of the proposed IDEA-QL-DL schemewas identified to incur only 6.56 and 6.39 seconds with respect to KDD'99 and DARPA datasets, which is comparatively lower than the reviewed works of the literature. In thenear future, it is planned to formulate different anomaly detection schemes by using the architecture of ResNets and AlexNets.

Declarations

Funding : Not applicable

Conflicts of interest/Competing interests: No Conflicts of interest/Competing interests

Availability of data and material: Data available on request due to privacy/ethical restrictions

Code availability: Code available on request due to privacy/ethical restrictions

References

1) Q. Cao, Y. Qiao and Z. Lyu, "Machine learning to detect anomalies in web log analysis," 2017 3rd IEEE International Conference on Computer and Communications (ICCC), Chengdu, 2017, pp. 519-523.
2) T. Jia, P. Chen, L. Yang, Y. Li, F. Meng and J. Xu, "An Approach for Anomaly Diagnosis Based on Hybrid Graph Model with Logs for Distributed Services," 2017 IEEE International Conference on Web Services (ICWS), Honolulu, HI, 2017, pp. 25-32.
3) X. Xia, W. Zhang and J. Jiang, "Ensemble Methods for Anomaly Detection Based on System Log," 2019 IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC), Kyoto, Japan, 2019, pp. 93-931.
4) A. Wadekar, T. Gupta, R. Vijan and F. Kazi, "Hybrid CAE-VAE for Unsupervised Anomaly Detection in Log File Systems," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 2019, pp. 1-7.
5) Y. Yuan, S. SrikantAdhatarao, M. Lin, Y. Yuan, Z. Liu and X. Fu, "ADA: Adaptive Deep Log Anomaly Detector," IEEE INFOCOM 2020 - IEEE Conference on Computer Communications, Toronto, ON, Canada, 2020, pp. 2449-2458.
6) N. Shone, T. N. Ngoc, V. D. Phai and Q. Shi, "A Deep Learning Approach to Network Intrusion Detection," in IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 2, no. 1, pp. 41-50, Feb. 2018.
7) H. Studiawan, F. Sohel and C. Payne, "Sentiment Analysis in a Forensic Timeline With Deep Learning," in IEEE Access, vol. 8, pp. 60664-60675, 2020.
8) Du, M., Li, F., Zheng, G., &Srikumar, V. (2017). DeepLog. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 3(2), 34-42.
9) R. Yang, D. Qu, Y. Gao, Y. Qian and Y. Tang, "nLSALog: An Anomaly Detection Framework for Log Sequence in Security Management," in IEEE Access, vol. 7, pp. 181152-181164, 2019
10) Z. Ghafoori, S. M. Erfani, J. C. Bezdek, S. Karunasekera and C. Leckie, "LN-SNE: Log-Normal Distributed Stochastic Neighbor Embedding for Anomaly Detection," in IEEE Transactions on Knowledge and Data Engineering, vol. 32, no. 4, pp. 815-820, 1 April 2020.
11) Z. Liu, T. Qin, X. Guan, H. Jiang and C. Wang, "An Integrated Method for Anomaly Detection From Massive System Logs," in IEEE Access, vol. 6, pp. 30602-30611, 2018

12) Wang, J., Tang, Y., He, S., Zhao, C., Sharma, P. K., Alfarraj, O., &Tolba, A. (2020). LogEvent2vec: Logevent-to-Vector based anomaly detection for large-scale logs in Internet of things. Sensors, 20(9), 2451.
13) Farzad, A., & Gulliver, T. A. (2020). Unsupervised log message anomaly detection. ICT Express, 6(3), 229-237.
14) M. A. Elsayed and M. Zulkernine, "PredictDeep: Security Analytics as a Service for Anomaly Detection and Prediction," in IEEE Access, vol. 8, pp. 45184-45197, 2020.
15) Wang, B., Ying, S., Cheng, G., Wang, R., Yang, Z., & Dong, B. (2020). Log-based anomaly detection with the improved k-nearest neighbor. International Journal of Software Engineering and Knowledge Engineering, 30(02), 239-262.

Figures

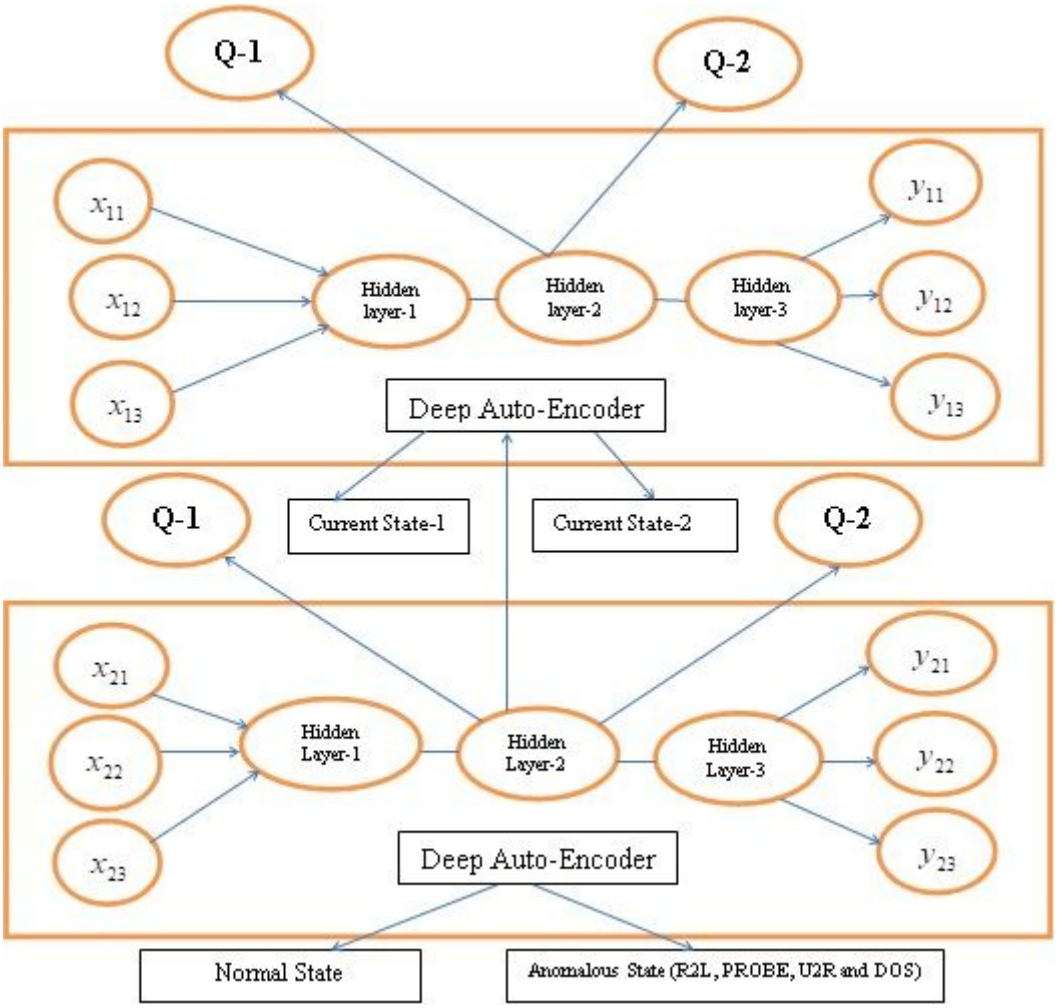

Figure 1

Architecture of DAEQN used for anomaly-based detection

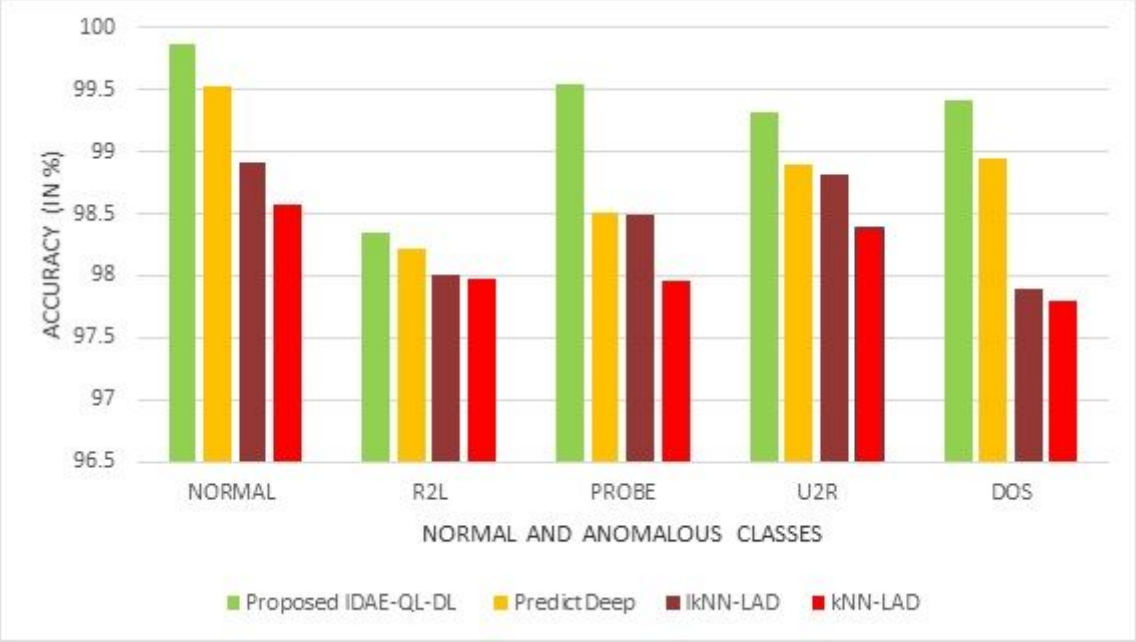

Figure 2

IDEA-QL-DLScheme-Accuracy in anomaly detection with KDD'99 dataset

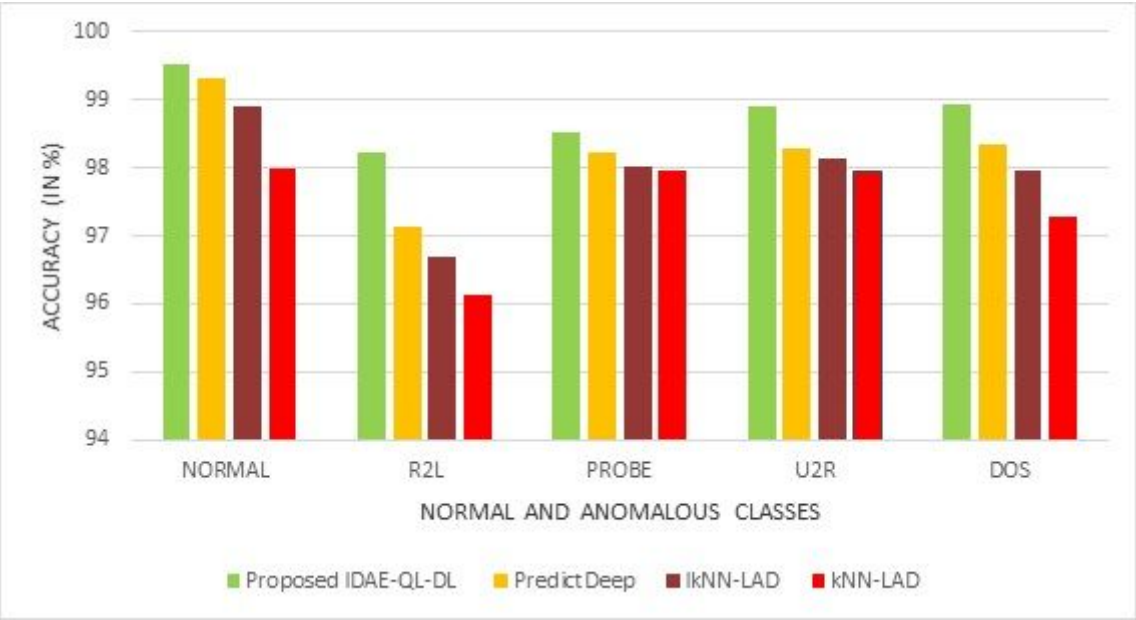

Figure 3

IDEA-QL-DLScheme-Accuracy in anomaly detection with DARPA'98 dataset

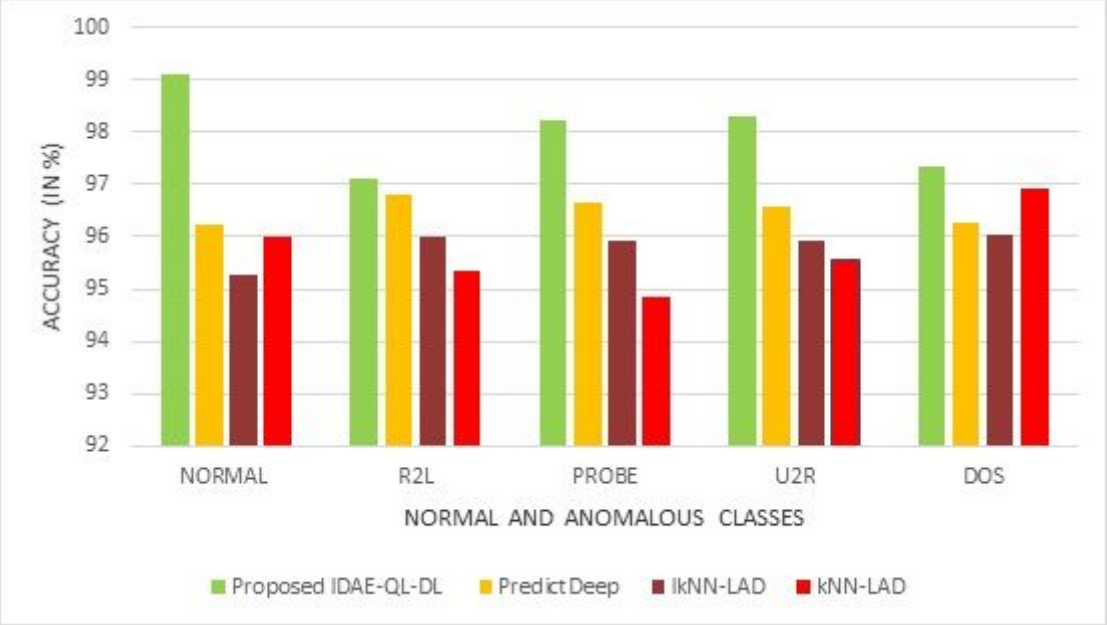

Figure 4

IDEA-QL-DLScheme-Accuracy in anomaly detection with synthesized dataset

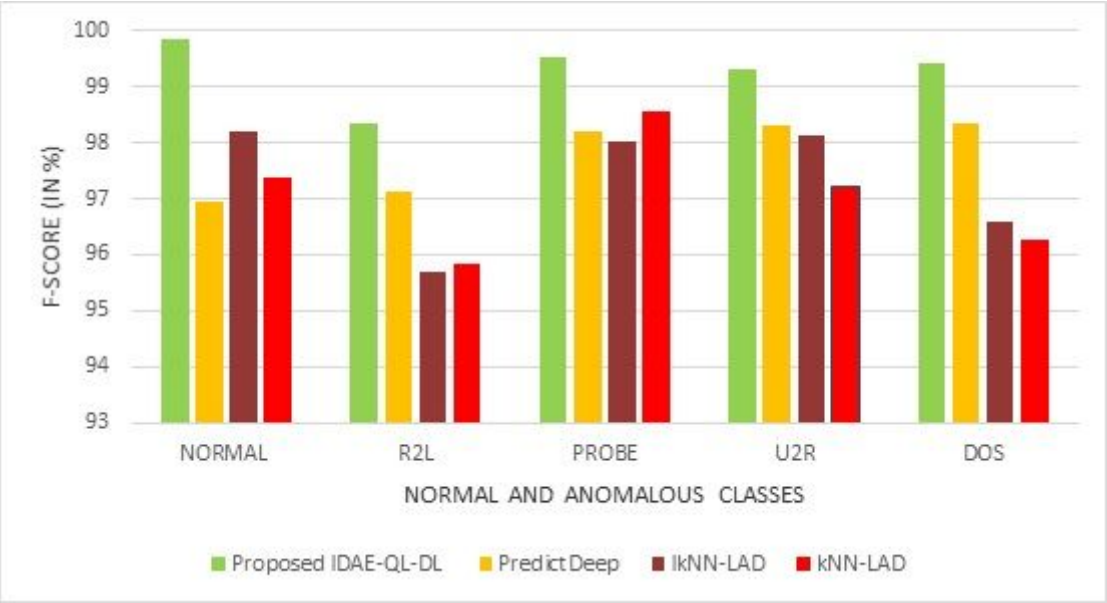

Figure 5

IDEA-QL-DLScheme-F-Score in anomaly detection with KDD’99 dataset

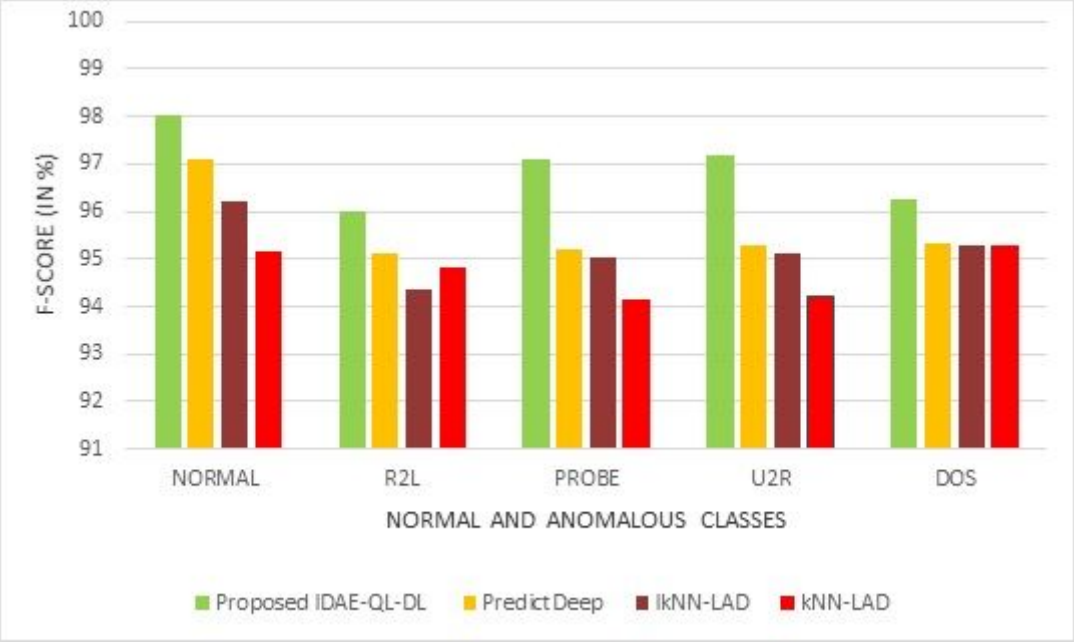

Figure 6

IDEA-QL-DLScheme-F-Score in anomaly detection with DARPA'98 dataset

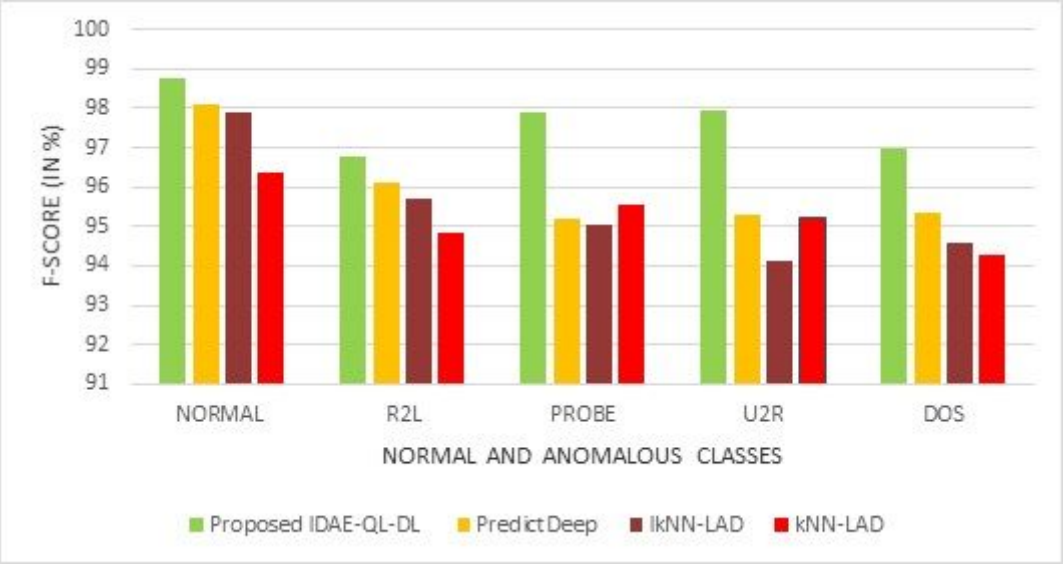

Figure 7

IDEA-QL-DLScheme-F-Score in anomaly detection with synthesized dataset

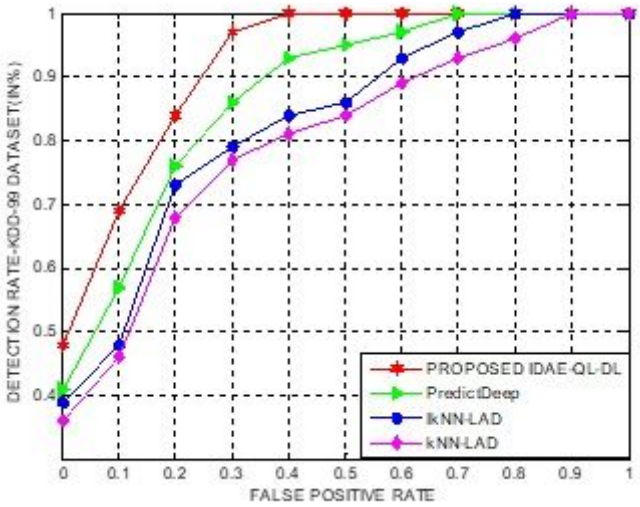

Figure 8

IDEA-QL-DLScheme-Anomaly Detection rate Vs. False positive rate with KDD’99 dataset

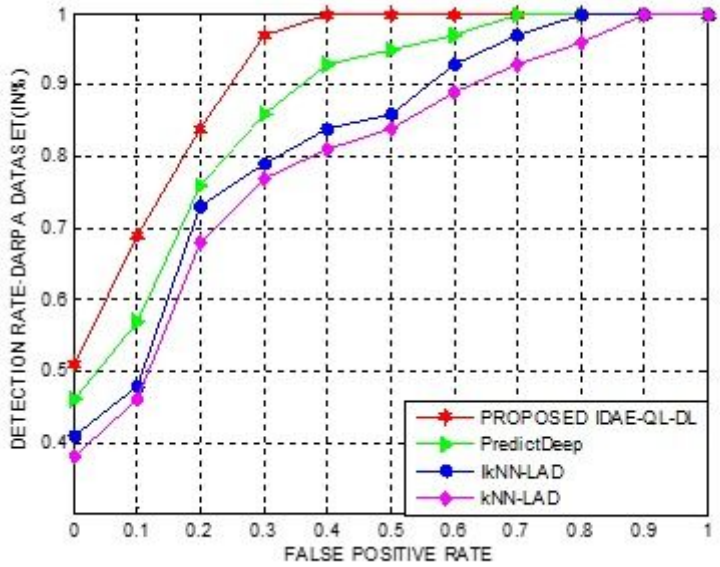

Figure 9

IDEA-QL-DLScheme-Anomaly Detection rate vs False positive rate with DARPA’98 dataset

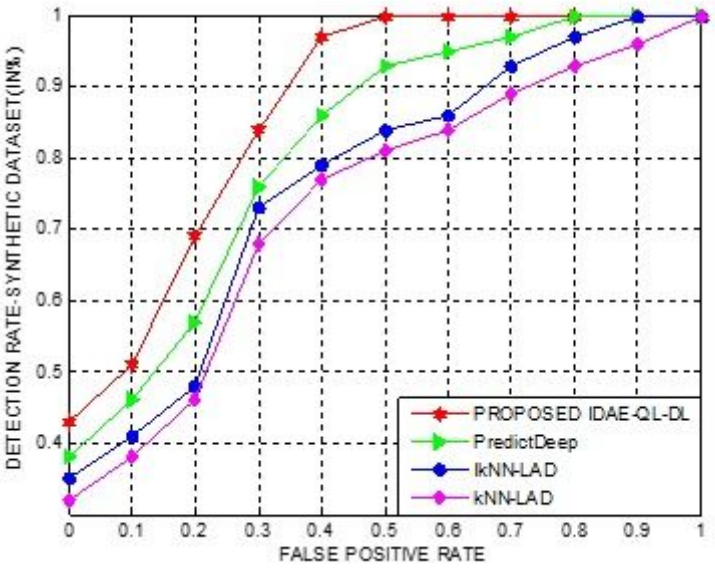

Figure 10

IDEA-QL-DLScheme-Anomaly Detection rate vs False positive rate with synthesized dataset