This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3030726, IEEE Access

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Deep Learning Enabled Data Offloading with Cyber Attack Detection Model in Mobile Edge Computing Systems

T. Gopalakrishnan[1], D. Ruby[1], Fadi Al-Turjman[2], Deepak Gupta[3], Member, IEEE, Irina V. Pustokhina[4], Denis A. Pustokhin[5] and K. Shankar[6], Member, IEEE

[1]Assistant Professor, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, 632014, India. (gopalakrishnan.ct@gmail.com) (ruby.d@vit.ac.in)
[2]Artificial Intelligence Dept., Research Center for AI and IoT, Near East University, Nicosia, 99138, Mersin 10, Turkey. (fadi.alturjman@neu.edu.tr)
[3]Department of Computer Science & Engineering, Maharaja Agrasen Institute of Technology, Delhi, 110086, India. (deepakgupta@mait.ac.in)
[4]Associate Professor, Department of Entrepreneurship and Logistics, Plekhanov Russian University of Economics, 117997 Moscow, Russia. (ivpustokhina@yandex.ru)
[5]Associate Professor, Department of Logistics, State University of Management, 109542 Moscow, Russia. (dpustokhin@yandex.ru)
[6]Department of Computer Applications, Alagappa University, Karaikudi, 630003, India. (drkshankar@ieee.org)

Corresponding author: K. Shankar (e-mail: drkshankar@ieee.org).

ABSTRACT Mobile edge computing (MEC) becomes popular as it offers cloud services and functionalities to the edge devices, to enhance the quality of service (QoS) of end-users by offloading their computationally intensive tasks. At the same time, the rise in the number of internet of things (IoT) objectives poses considerable cybersecurity issues owing to the latest rise in the existence of attacks. Presently, the development of deep learning and hardware technologies offers a way to detect the present traffic condition, data offloading, and cyber-attacks in edge networks. The utilization of DL models finds helpful in several domains in which the MEC provides the decisive beneficiary of the approach for traffic prediction and attack detection since a large quantity of data generated by IoT devices enables deep models to learn better than shallow approaches. In this view, this paper presents a new DL based traffic prediction with a data offloading mechanism with cyber-attack detection (DLTPDO-CD) technique. The proposed model involves three major processes traffic prediction, data offloading, and attack detection. Initially, bidirectional long short term memory (BiLSTM) based traffic prediction to enable the proficient data offloading process. Then, the adaptive sampling cross entropy (ASCE) technique is executed to maximize the network throughput by making decisions related to offloading users to the WiFi system. Finally, a deep belief network (DBN) optimized by a barnacles mating optimizer (BMO) algorithm called BMO-DBN is applied as a detection tool for cyberattacks in MEC. Extensive simulation is carried out to ensure the proficient performance of the DLTPDO-CD model. The experimental outcome stated the superiority of the presented model over the compared methods under different dimensions.

INDEX TERMS Mobile edge computing, Data Offloading, Traffic Prediction, Security, Deep Learning.

I. INTRODUCTION

Internet of Things (IoT) is a heterogeneous resource-based environment that is applicable to offer seamless services for satisfying the end-user requirements. The concept of IoT differs from tiny sensors that are comprised of communication, as well as processing features with memory space. The intelligence of these machines and human communicative behavior is exploited to facilitate user demands [1]. The units developing "things" visualize details in electronic format for machine-level learning. The heterogeneous integration of the IoT platform is expanded for interaction technique and embedding Long Term Evolution (LTE), Short-Range Communications (SRC), Wi-Fi, WI-Max methodologies, and so forth. It guides the individuals for demanding and distributing multi-level information data which can be shared through the system [2]. In particular, the communication model offers persistent as well as ubiquitous permission for end-users to record and retrieve the

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

information. The shared data sources like cloud, data centers, decentralized sectors, and services are combined with the application of autonomous connectivity features of the network.

Using the extensive application of several models and communication tools, the IoT concept has been utilized in the medical sector, military services, industrial fields, mobile communications, and so on. The amount of data managed by IoT cannot be defined specifically because of the random memory space, request rate as well as the density of a device. Since a persistent communication platform is provided, data management and flow differ with respect to time. Massive amount of data storage and allocation facilities can be offered for IoT users by making simple requests flow and responses through end-to-end servers and gateways [3]. The merits of the cloud are data processing, offloading, shared access, and many other factors that are inherited by IoT units and reconcile tools to evade the effect of data flows, overcrowding, and backlogs. The additional features are maintained for the IoT environment by the dedicated edge and fog computing structures. Such features are generated to assure fewer features as well as better communication.

Edge computing is an alternate communication model that offers services to dedicated users in the communicating system. Such tools are facilitating as mobile computing models with autonomous servers and functions which are obtained from cloud or IoT layers. The derived resources are expanded to users for providing difficult services [4]. The facilities like support transformation, mobility, adaptability, elasticity, and reliable features that support varying user device density. The request computation is streamlined from the edge layers to top-layers like cloud or IoT. Various facilities and applications associate edge and fog structures are considered as the portion of IoT model for generating scalable communication platform with data analytics as well as computation [5]. The facilities offered and inherited in upper layers are segmented over accessible users in a system for approving the non-colliding responses for demanded resources. Specifically, request handling, data distribution and system management, user responses, delay-less services, mediate accessibility, and so forth, are considered as the advantages of the processing approaches in the large-scale system like cloud and IoT.

The major problem is reported by Human-Driven Edge Computing defines how to loop users. The huge proliferation of personal computing tools is opening novel human-based designs which reduce the edges from human and device. Additionally, edge facilities have been employed for interchanging the data gathered and computed inside the content of IoT towards physical services and for visualizing conventional search engines by individuals. Then, former studies on data control depend upon related to edge processing which is comprised of a structure with massive collaborative multitudes of processing nodes that are fixed among sensor nodes and cloud-related facilities. These mediating levels are applied for performing substantial data storage and computation for reducing the retrieving duration interms of cloud-relied services, and to apply the least resources as well as power for diminishing the workload.

The interdependencies between these 3 diverse phases of memory and processing inside an IoT solution are tedious, and computing data collection is a challenging issue which is highly impossible. At this point, while considering the superiority of objectives like consistency, accessibility, trust, mobility, as well as power efficiency, when actual behavior of a network and service duration of the device's batteries. The examination of human action and communication with external as well as digital noises are applicable in closing the control loop of adaptive distributed networks. It is a new research environment for distributed systems that applies user behaviors under diverse contents, shifting massive network edge by devices like the fifth generation (5G) mobile networks.

The effective way of preserving energy as well as reducing delay, caching resources were examined under various applications. For enabling users for implementing delay-sensitive as well as context-aware sectors, the Mobile Edge Computing (MEC) servers are applied in Small Base Stations (SBS). Hence, system efficiency has been enhanced by applying the caching and processing abilities of MEC. In [6], Zhang et al. have presented an energy effective processing offloading method which considers front-haul and back-haul links, that desires for optimizing offloading energy for operations at the constraining the delay. Wang et al. [7] together optimize the power transmission beamforming at the access point (AP) and resource assignment between users for enhancing MEC function. Guo et al. [8] establish a power-efficient offloading principle for multi-access edge computing across hybrid fiber-wireless networks. Moreover, Khanna et al. [9] apply MEC to compute offloading for accomplishing energy preservation. Thus, MEC has been applied for resolving problems experienced in novel industries such as IoT. Zhang et al. [10] project a cloud-aided MEC offloading model in a vehicular system for enhancing the function of vehicular services. Yang et al. [11] present a communications-based MEC mechanism for reducing the communication-resource application of VR models.

In order to develop mobile data and resolve the constraints of licensed resources, mobile data offloading reduces the overhead of high-loaded SBSs has to be assumed. Mobile data offloading methods are divided into 4 classes namely, data offloading by small cell systems, WiFi networks, opportunistic mobile networks as well as heterogeneous systems. Researchers of [12] have developed an uplink traffic offloading method according to dual-connectivity, that enables mobile units flexibly modules for traffic among a macro-cell Base Station (BS) and a small cell AP. Moreover [13] concentrates on offloading traffic on overhead SBS to small cell networks. Moreover, offloading users to WiFi networks gained massive concentration in business and educational sectors as it is an inexpensive and efficient model for accomplishing load balancing in this network.

Feng et al. [14] developed data traffic that is offloaded to WiFi and model the best principle that assumes user satisfaction as well as benefit losses. A comprehensive study on offloading cellular traffic for the WiFi system is projected in Lee et al. [15], which showcases the substantial development is obtained in case of massive data transmission. To encourage the WiFi system in traffic offloading, Zhang et al. [16] presented quality of service (QoS) aided incentive approach and gained massive benefits under minimum cost. To eliminate ineffective function because of additional handover to WiFi AP, Jang, and Chang [17] apply the Software-defined networking (SDN) method for managing handover between WiFi AP and BS, where the network is applicable for offering huge service data rate. By assuming the complicated congestion, numerous users' requests are derived which can be resolved by [18] a method named mobile participation that guides data offloading, and pre-caches massive context over congested areas. Besides, mobile data offloading issues of MEC is highly focused by many developers.

Presently, the advent of deep learning (DL) models paves a way in the prediction of traffic loads and attack detection to ensure security in MEC. The application of DL models finds beneficial in several fields. This paper presents a new DL based traffic prediction with a data offloading mechanism with cyber attack detection (DLTPDO-CD) technique for MEC. The proposed model involves three major processes traffic prediction, data offloading, and attack detection. Firstly, bidirectional long short term memory (BiLSTM) based traffic prediction to enable the proficient data offloading process. Once the traffic load is predicted, data offloading process takes place using adaptive sampling cross entropy (ASCE) technique to maximize the network throughput by making decisions related to offloading users to the WiFi system. Finally, deep belief network (DBN) optimized by barnacles mating optimizer (BMO) algorithm called BMO-DBN is applied as a detection tool for cyberattacks in MEC. Extensive simulation is carried out to ensure the proficient performance of the DLTPDO-CD model. The experimental outcome stated the superiority of the presented model over the compared methods under different dimensions.

II. THE PROPOSED MODEL

The MEC-related structure comprises SBS, WiFi AP, MEC, and Evolved Packet Core (EPC) is co-existed. Users' transmissions as well as processing delay are reduced significantly by processing tasks onto MEC. Under the inexistence of MEC servers, users will be able to retrieve the data directly from EPC, which fails for satisfying the user demands. Intuitively, the case of EPC has been assumed, where SBS, WiFi AP, and N users apply diverse details of this study. Followed by, the MEC servers are placed in the proximity of every SBS and WiFi AP, thus the contents have been employed. It refers that, if SBS or WiFi AP offers data, the communication delay could be limited by directly providing the cached details, and it is applicable to satisfy the demands of wireless users.

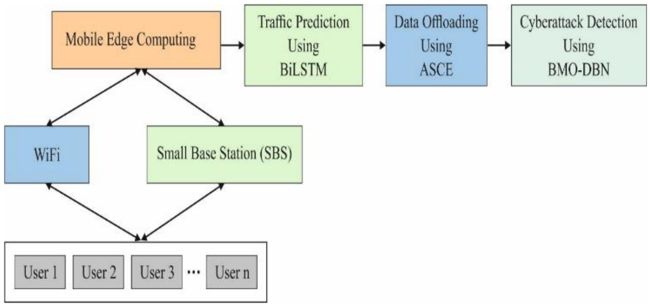

FIGURE 1. The overall work of the proposed model.

The contents are classified into K classes on the basis of the data transmission rate as well as the priorities. Users with lower data rate demands such as low-priority users will have the least application of SBS resource which has been offloaded to the WiFi AP and SBS traffic goes beyond the threshold. In addition, the resource application of SBS and WiFi systems have been enhanced which results in system throughput. Fig. 1 shows the working process of the BMO-DBN model.

A. Bi-LSTM based Traffic Prediction

Mobile data offloading is considered as challenging issues of improving the entire system throughput which ensures the users' data rate demands and reduces the overhead of traffic SBS. Firstly, contents are classified as categories on the basis of the data transmission rate as well as delay requirements. Moreover, DL method is applied for predicting the SBS traffic at the subsequent moment. Users accessing the content of low data broadcast rate necessities are assumed to contain the least priority that is described as capable offloaded objects when the SBS is overhead. Consequently, the users having lower offload have attained overall throughput of the system, meanwhile it is considered with the permit condition of end-user as well as traffic overload of SBS.

Typically, a recurrent neural network (RNN) is composed of complexities in learning long-term dependencies [19]. An LSTM-based method is extended to RNNs that are capable to deal with diminishing gradient problem. It expands the RNNs' memory for enabling prominent long-term dependencies of inputs. These memory extensions have the capability of memorizing data more a longer duration and activate read, write, and remove data from the memories. An LSTM memory is known as "gated" cell, but the word gate is simulated for making the decision of adding or removing the memory data. The LSTM method acquires essential features from inputs and maintains the data for the massive time duration. A decision of removing or maintaining the data depends on the weight values allocated to the data in the training procedure. Thus, the LSTM method learns the data significance for preserving or removing. Generally, an LSTM method contains 3 gates namely forget, input, and output gates. In forget gate, the decision of maintaining or deleting the predefined data takes place; secondly, an input gate has coverage for novel data which is recorded into the memory, finally, output gate manages a predefining value in the cell and gives the outcome.

Forget Gate. The sigmoid function is generally utilized for making a decision that requires to be deleted from the LSTM memory. These decisions are fundamentally based on the value of h_{t-1} and x_t. An outcome of this gate is f_t, a value among 0 as well as 1, whereas 0 denotes the learned value, and 1 indicates the entire value. The outcome is calculated as:

$$f_t = \sigma(W_{f_h}[h_{t-1}], W_{f_x}[x_t], b_f) \tag{1}$$

where b_f is a constant and is known as bias value.

Input Gate. Here, it develops the decision of whether or not the novel data is within the LSTM memory. This gate is composed of 2 layers: sigmoid layer and "tanh" layer. A sigmoid layer generates the values that have to be updated, and tanh layer generates the vector of novel candidate values which is further provided into LSTM memory. The outcomes of 2 layers are calculated as:

$$i_t = \sigma\left(W_{i_h}[h_{t-1}], W_{i_x}[x_t], b_i\right) \tag{2}$$

$$c_t = \tanh\left(W_{c_h}[h_{t-1}], W_{c_x}[x_t], b_c\right) \tag{3}$$

where i_t implies whether the value requires to be updated or not, and c_t denotes a vector of novel candidate values which is further into the LSTM memory. A group of these 2 layers gives an update to LSTM memory in that present value is forgotten utilizing the forget gate layer with the multiplication of the old value $(i.e., c_{t-1})$ afterward further a novel candidate value $i_t * c_t$. The following equation signifies its mathematical equation:

$$c_t = f_t * c_{t-1} + i_t * c_t \tag{4}$$

where f_t is the outcomes of the forget gate that the value among 0 as well as 1 where 0 represents finally obtain rid of the value; but, 1 denotes finally maintain the value.

Output Gate. This gate utilizes the sigmoid layer for making the decision that is a division of LSTM memory gives to the outcome. After that, it makes a non-linear tanh function for mapping the values among -1 and 1. At last, an outcome is multiplied from the sigmoid layer. The following equation signifies the formulas for calculating the outcome:

$$o_t = \sigma\left(W_{o_h}[h_{t-1}], W_{o_x}[x_t], b_o\right) \tag{5}$$

$$h_t = o_t * \tanh(c_t) \tag{6}$$

where o_t is the outcome value, and h_t is illustrated as a value among -1 and 1.

1) DEEP BIDIRECTIONAL LSTMS (BILSTM)

A deep-bidirectional LSTMs are further explained, in that 2 LSTMs are executed to the input data [19]. During the initial round, the LSTM is implemented on the input series (i.e., forward layer). In the second round, a reverse type of the input series is provided into the LSTM method (i.e., backward layer). Executing an LSTM double is developed to enhance learning long-term dependencies and it enhances the accuracy of a method. Fig. 2 illustrates the structure of the Bi-LSTM model.

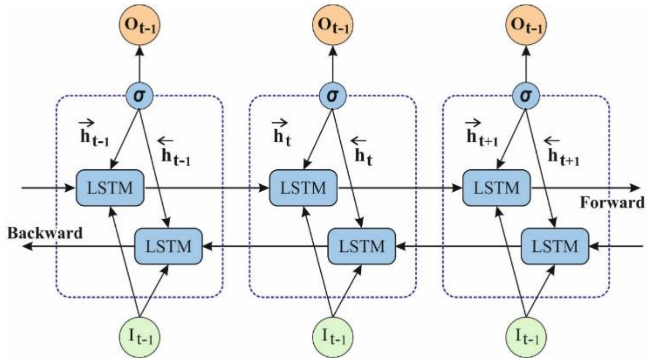

FIGURE 2. The Structure of Bi-LSTM.

B. Adaptive Sampling Cross Entropy based Data Offloading

It is one of the binary integer programming issues that can be resolved effectively through the branch-and-bound (BnB) model with massive processing complexity, in particular, if X is higher. Moreover, a huge count of operations would enhance the computational access points (CAPs) method. Followed by, BnB technology meets the demands of practical applications. Followed by, the works that attempt to resolve the issues by applying traditional optimization approaches. A well-known result is to apply convex relaxation, such as relax $x_{nm} \in \{0,1\}$ as $x_{nm} \in [0,1]$ by linear programming relaxation (LPr) [20]:

$$T(X) = \max_{m \in \mathcal{M}} T_m(X) \ as \ T(X) \geq \max_{m \in \mathcal{M}} T_m(X) \tag{7}$$

by semi-define relaxation (SDR). However, relaxation tends to degrade the performance when compared to the BnB approach.

Followed by, the discrete optimization parameters are resolved by applying a probabilistic method in learning the probability of a policy x_{nm}. The CE method is defined as a probability learning scheme in machine learning (ML) application. To resolve the problem, CE technology has been applied along with adaptive sampling like ASCE model.

1. THE CROSS ENTROPY (CE) CONCEPT

Here, CE is also named as probability theory as Kullback-Leibler (K-L) divergence is facilitated as a measure of distance among 2 probability distributions. In case of 2 distributions, $q(x)$ and $p(x)$, CE is described as follows [20]:

$$D(q||p) = \underbrace{\sum q(x) lnq(x)}_{H(q)} - \underbrace{\sum q(x) lnp(x)}_{H(q,p)}. \tag{8}$$

It is pointed that, newly presented CE-based learning scheme $p(x)$ shows a theoretically-tractable distribution method that attempts to learn and achieve the best results, whereas $q(x)$ denotes the empirical distribution that simplifies true distribution of better solutions. In particular, in ML, distribution $q(x)$ is learned from observations and $H(q)$ denotes the entropy of $q(x)$, which results in the similarity of learning CE in (8) and $H(q,p)$.

From the description of CE, a well-known cost function in ML, and resolve issues through probability learning. It is learned from $p(x)$ that iteratively training instances are generated for achieving optimal policy of X on the basis of $p(x)$, which is similar to empirical solution, $q(x)$.

2. THE ASCE- BASED OFFLOAD LEARNING

In the case of probability learning, probability distribution function $p(x)$ has been applied with a descriptor u, such as $p(x, u)$ could be a Gaussian distribution as well as u is composed of mean and variance. It is pointed that L is equal to $N \times (M+1)$, the indicator u implies a vector of L dimensions, which is described as $u = [u_1, u_2, \cdots, u_L] \triangleq [u_0^T, u_1^T, \cdots, u_M^T]$ where $u_m^T = [u_{1m}, u_{2m}, \cdots, u_{Nm}]$ and $u_{nm} \in [0,1]$ indicates the probability of $Pr(x_{nm} = 1)$. Under the application of this approach, it is known $p(x; u)$ by learning the parameter u. Additionally, X is vectorized as $x = [x_0^T, x_1^T, \cdots, x_M^T]$ in which $x_m^T = [x_{1m}, x_{2m}, \cdots, x_{Nm}]$. Under the application of Bernoulli distribution, the distribution function $p(x, u)$ is formalized as:

$$p(x, u) = \prod_{l=1}^{L} u_l^{x_l} (1 - u_l)^{(1-x_l)}. \quad (9)$$

Based on (8), an initial process correlates to top CAP. Hence, when an operation is allocated to CAP then the probability of being associated with alternate CAPs is 0. For reducing the repetition of produced instances, a single sample is divided into, a vector x of L dimensions, into $M+1$ autonomous blocks, $x_0, x_1, \cdots, x_M$, and it is associated with CPU, $[x_{1m}, \cdots, x_{Nm}]$ refers to the operation assignment of 1-N to CAP m. Assume $\mathcal{G}$ implies the collection of indices of decided blocks in sampling and $\mathcal{T}$ is an alternate set for saving the instances to meet the limitations for all iterations. Initially, select g, an index in $M \setminus \mathcal{G}$. For generating an x_g given g, the entries of x_g based on the probability density function $p(x_g, u)$. For all u_l in x_g, it is derived based on Bernoulli distribution of parameter u_l. The indicator u_m of residual blocks in $M \setminus \mathcal{G}$ is modified based on x_g, such that, when $x_{ig} = 1$ then $u_{im} = 0$ for $m \in M \setminus \mathcal{G}$. If the cardinality of $\mathcal{G}$, is implied as $|\mathcal{G}|$, is the same as M, as the valuable sample has been generated. The valuable samples are collected in $\mathcal{T}$ and sampling is followed by the cardinality of $\mathcal{T}$, it is implied as $|\mathcal{T}|$, and attained S.

In the presented CE model, processing in all iteration is performed in parallel, while iterations are executed prominently. From the attained simulation outcome, the hyperparameters are adjusted in the presented model, like S, S_{elite}, α. It makes a simple trade-off among performance and delay. The CE in Eq. (8) has been applied as a lost function. It depicts that minimum $H(q, p)$ is, the lower the distance from $q(x)$ and $p(x)$. It denotes that,

$$\min H(q, p) = \max \sum q(x) \ln p(x)$$
$$= \max \frac{1}{S} \sum \ln p(x, u), \quad (10)$$

where $q(x)$ is $\frac{1}{S}$, as the probability of autonomous result in a group of samples are $1/S$ where S denotes the cardinality of a set. Based on the issues in (10), the objective is used for identifying the best optimal indicator u for reducing $H(q, p)$. In tth iteration, S series of random instances x, is allocated as a candidate that is retrieved based on the probability $p(x, u)$. The possible sample produced by adaptive sampling is determined. It is evaluated that the objective $\{\Psi(x^s)\}_{s=1}^{S}$ is calculated and it is organized as $\Psi(x^{[1]}) \le \Psi(x^{[2]}) \le \cdots \le \Psi(x^{[S]})$.

Followed by, S_{elite} samples, such as, $x^{[1]}, x^{[2]}, \cdots, x^{[elite]}$, which offers the lower objective, are chosen as elites. Then, optimal indicator u for strategy x is computed as:

$$u^* = \arg \max_{\mathrm{u}} \frac{1}{S} \sum_{s=1}^{S_{e1ite}} \ln p(x^{[s]}, u) \quad (11)$$

By applying (9) and (11) and by encouraging $\frac{\partial H(q,p)}{\partial u_l} = 0$, the saddle point c

$$u_l^* = \frac{1}{S_{e1ite}} \sum_{s=1}^{S_{e1ite}} x_l^{[s]}. \quad (12)$$

From the presented model, the CE-related measure is used for expanding the probability. By assuming the randomness of sampling, count of samples are minimum, the function $u^{(t+1)}$ is upgraded in $(t+1)th$ iteration based on u^* that is managed using (11) and (12), and $u^{(t)}$ is learned in the final iteration. It applies

$$u^{(t+1)} = \alpha u^* + (1 - \alpha) u^{(t)}, \quad (13)$$

Where $\alpha \in [0, 1]$ implies the learning value. Generally, for CE-aided model the iterations converging the best optimized solution of issues.

C. CyberAttack Detection using DBN model

Once the traffic load is predicted and data offloaded, the DBN model is applied for the detection of cyberattacks in MEC. DBN is a typical deep neural network (DNN) method that is comprised of many restricted Boltzmann machines (RBMs) and a classification layer. Standard DNN contains unsupervised pre-training of deep RBMs and supervised fine-tuning of the classifier layer. The DBN illustrates optimal feature extraction and it is applicable in feature learning. In DBN, the fully connected (FC) network is utilized for examining data when compared with normal DNNs (e.g., convolution neural network (CNN), RNN). It allows the presented technique for extracting data and provide it into DBN for working better in equipment error analysis. So, DBN is utilized as an important device to gearbox error analysis.

As illustrated in Fig. 3, all RBMs makes a visible layer with visible units $v = \{v_1, v_2, \ldots, v_i\}$, and hidden layer that includes the hidden units $h = \{h_1, h_2, \ldots, h_j\}$. The model parameters of DBN $\theta = [W, b, a]$, the energy function is provided as [21]:

$$E(v, h; \theta) = -\sum_{i=1}^{I} \sum_{j=1}^{J} \omega_{ij} v_i h_j - \sum_{i=1}^{I} b_i v_i$$
$$- \sum_{j=1}^{J} a_j h_j \quad (14)$$

where ω_{ij} refers to the link weight among visible unit v_i and the entire number is I as well as hidden unit h its entire number is J, b_i and a_j indicate the bias conditions of the visible as well as hidden units, correspondingly. A combined sharing entire unit is computed depends on the energy function $E(v,h;\theta)$ as:

$$p(v,h;\theta)=\frac{exp\left(-E(v,h;\theta)\right)}{Z} \tag{15}$$

where $Z=\sum_{h;v} exp\left(-E(v,h;\theta)\right)$ is the separated function. The conditional possibilities of the hidden as well as visible units h and v is computed as follows:

$$p(h_i=1|v;\theta)=\delta\left(\sum_{i=1}^{I}\omega_{ij}\,v_i+a_j\right) \tag{16}$$

$$p(v_i=1|v;\theta)=\delta\left(\sum_{j=1}^{J}\omega_{ij}\,h_i+b_i\right) \tag{17}$$

where δ is a logistic function, i.e., $\delta(x)=1/1+\exp(x)$. The RBMs are trained for maximizing the combined possibility. DBN is created by stacking several RBMs, where the outcome of the lth layer is utilized as the input of the $l+1th$ layer. In the training model, DBN is simplified into 2 phases such as pre-training and fine-tuning. Initially, the information is sustaining with a visible layer of initial RBM and changed into the hidden layer that is frequently applied in RBM. Then, layer-to-layer unsupervised training has been completed, and features discovered automatically by DBN are sustained with the classification layer of DBN. At last, fine-tuning is executed in the classification layer for improving DBN. Finally, the softmax layer is used for classification purposes.

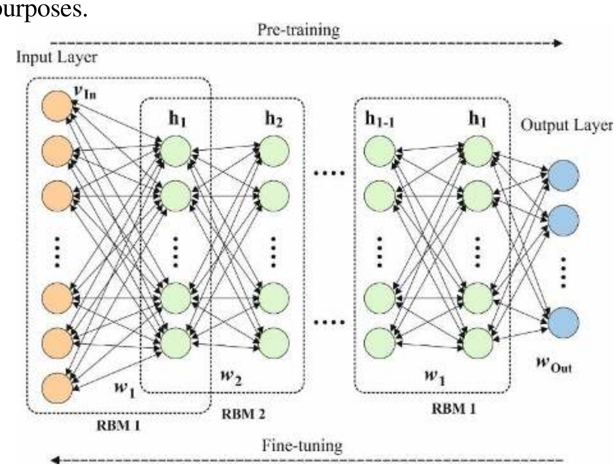

FIGURE 3. The Architecture of DBN.

1. LEARNING RATE SCHEDULER USING BMO ALGORITHM

The barnacles are micro-organisms, which are presented as Jurassic times [22]. A barnacle can swim from their births and if it attains the adult stage, it attaches themselves to objects in the water and grows a shell. An important interesting fact on barnacles is they are well-known to their extended penises, containing a few longest in animals connect to their body that is 7 to 8 times the length of their bodies for managing with the altered tides and sedentary way of life. Another unique feature of barnacles is their ability to tenacious underwater adhesion by secreting proteinaceous substances. But, in this paper, the mating procedure of barnacles is utilized as a simulation of the BMO algorithm [22].

1) Hardy−Weinberg principle

This section presents a new BMO algorithm to tune the hyperparameters of the DBN model. The Hardy-Weinberg principle would be applied in producing off-springs of BMO. Simply, the 2 alleles of D and Mdenotes the Dad and Mum with frequencies$f(D)=p$ and $f(M)=q$, correspondingly, the desired genotype frequencies by mating which is showcased by $f(DD)=p^2$ for DD homozygotes, $F(MM)=q^2$ for MM homozygotes as well as $f(DM)=2pq$ to heterozygotes. The alternate method for developing the genotypes for new off-spring generation is expressed by Punnett square. It is viewed as $p=0.6, q=0.4$, that provides the region of a rectangle which shows genotype frequencies in the following: DD: DM : $MM=0.36; 0.48: 0.16$. The summation of entries is represented as $p^2+2pq+q^2=1$. Moreover, it is pointed out as $p+q=1$. Therefore, the generation of novel off-spring depends upon the p and q of barnacles' parents as given in the following.

2) BMO algorithm

The processes involved in the BMO algorithm are discussed as follows [22].

i. Initialization

Here, it is considered that candidate of the result is a barnacle while vector of the population could be depicted as:

$$X=\begin{bmatrix} x_1^1 & \cdots & x_1^N \\ \vdots & \ddots & \vdots \\ x_n^1 & \cdots & x_n^N \end{bmatrix} \tag{18}$$

where N refers to the count of controlling parameters and n shows the population count. The control variable in Eq. (18) is subjected to upper as well as lower bounds of the problem which has to be resolved in the following:

$$llb=[llb_1,\dots,llb_i] \tag{19}$$

$$lb=[lb_1,\dots,lb_i] \tag{20}$$

where ub and lb are the upper as well as lower bounds of ith variable. An estimate of the vector X is complete in the beginning, and the sorting processes have been carried out for placing the optimal solution at the top of the vector X.

ii. Selection process

A projected BMO algorithm has been applied with a method for selection which is related to EA models, as the selection of 2 barnacles is depends on the length of the penises, pl. The election procedure reflects the nature of barnacles that depends upon the provided considerations:

- The selection process is performed in a random manner; however, it is limited to the penis length of a barnacle pl.
- The election process chooses a similar barnacle that refers that self-mating is performed. Self-mating is highly random and barnacle male as well as female reproductions, hence, self-mating is assumed, and novel off-spring has been produced.
- When a selection is based on specific iteration which is higher than pl, the sperm casting is performed.

IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

It is pointed that from the earlier considerations that the task of exploitation (points no. 1 and 2) and exploration (points no 4) are enlisted in the algorithm. It is assumed as an optimal solution which is placed at the top of candidates of result X. Assume that the higher penis length of barnacles is 7 times maximum than size (pl =7), therefore at specific iteration, barnacle #1 is applicable for mating with barnacles #2–#7. When barnacle #1 chooses barnacle #8, it exceeds the restriction, hence the moderate mating is not performed. Thus, the offspring production is computed by the sperm cast process that is defined on the following. It is the way of this model which is carried out with respect to virtual distance and. The given simple selection has been applied and it is expressed in mathematical form:

$$barnacle\_d = randperm\ (n) \tag{21}$$

$$barnacle\_m = randperm\ (n) \tag{22}$$

where the $barnacle\_d$ and $barnacle\_m$ are defined as parents which have to be mated and $n$implies the count of the population. Eqs. (21) and (22) illustrates the chosen is developed arbitrarily and satisfies the consideration value 1 in the existing subsection.

iii. Reproduction

The reproduction procedure presented in BMO is related to EAs. Since there are no particular formulas for deriving the reproduction of barnacles, the BMO is simplified on inheritance genotype frequencies of barnacles' parents in generating the offspring relied on Hardy–Weinberg strategy. For representing the simplicity of projected BMO, the following function is presented for developing novel variables of offspring from barnacles' parents:

$$x_i^{N\_new} = px_{barnacle\_d}^{N} + qx_{barnacle\_m}^{N} \tag{23}$$

where p defined the scattered normally with pseudo-random values from $[0,1]$, $q = (1 - p)$, $x_{barnacle\_d}^{N}$ and $x_{barnaclem}^{N}$ are defined as variables of Dad and Mum of barnacles correspondingly. It is decided in Eqs. (21) and (22). Therefore, the offspring inherits the natures of Dad and Mum according to the possibility of random values within 0 to 1. For simple execution, let p be 0.6 which refers that 60% of Dad's features and 40% of Mum's features are incorporated in the novel offspring generation.

It can be necessary to point a measure of pl which acts as a significant one in calculating the exploitation as well as exploration operations. When the chosen barnacles have to be mated in a specific range of penis length of Dad's barnacle, the exploitation task is taken place in (Eq. (23)). Sperm cast is performed while the chosen barnacles have to be processed and pl is allocated initially.

$$x_i^{n\_new} = rand(\quad) \times x_{barnacle\_m}^{n} \tag{24}$$

where $rand\ ()$ is the random number among $[0,1]$. It is pointed out that Eq. (24) displays an elegant method of barnacles offspring. The novel offspring is produced from Mum's barnacle for exploration. It is because of the new offspring is produced by Mum's barnacle as it obtains the sperm from water which is released by alternate barnacles.

The BMO initializes the optimization model by developing a collection of random solutions. Novel offspring of barnacles are produced according to Eqs. (23)–(24). An optimal result which has been upgraded in iteration where it can be placed at top of the vector X. For the management of matrix expansion from population size, a novel offspring of barnacles are determined and combined with the parents. Then, the sorting task is processed for selection of top solution which fits the population size. The worst outcomes are considered as dead or removed. Fig. 4 depicts the flowchart of the BMO method.

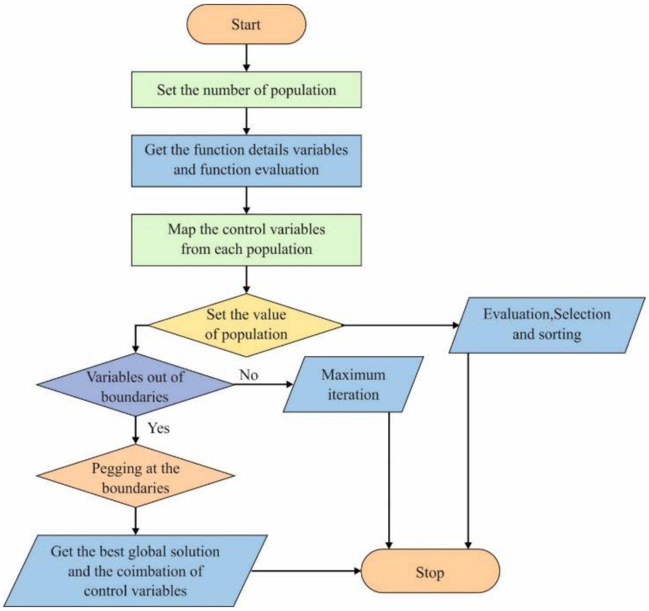

FIGURE 4. The Flowchart of BMO algorithm.

III. PERFORMANCE VALIDATION

This section discusses the performance of the proposed models under diverse aspects. Firstly, the traffic load predictive performance of the Bi-LSTM model is examined interms of NMSE under the varying geographical area. Table 1 and Fig. 5 display the NMSE values attained by the LSTM and Bi-LSTM models. The experimental values notified that the Bi-LSTM model has attained effective prediction results with the minimal NMSE compared to the LSTM model. For instance, under the geographical area of 5sq.km, the Bi-LSTM model has reached a lower NMSE of 0.011046 whereas the LSTM model has attained worse results with the higher NMSE of 0.013414. In line with this, under the geographical region of 10sq.km, the Bi-LSTM method has attained the least NMSE of 0.004350 while the LSTM approach has achieved inferior outcomes with the maximum NMSE of 0.006075.

Followed by, under the geographical area of 15sq.km, the Bi-LSTM technique has obtained a minimal NMSE of 0.002659 while the LSTM technology has accomplished poor simulation outcomes with greater NMSE of 0.004147. Meantime, under the geographical region of 20sq.km, the Bi-LSTM approach has obtained least NMSE of 0.001847 and the LSTM framework has reached poor outcomes with the greater NMSE of 0.003640. In addition, under the geographical area of 25sq.km, the Bi-LSTM scheme has accomplished a minimum NMSE of 0.000799 while the LSTM technique has reached inferior results with better NMSE of 0.001948. In continuing with, under the geographical region of 30sq.km, the Bi-LSTM scheme has achieved the least NMSE of 0.000257 while the LSTM scheme has reached poor results with maximum NMSE of 0.000799. Moreover, under the geographical region of 35sq.km, the Bi-LSTM technology has accomplished the least NMSE of 0.000224 while the LSTM technology has gained ineffective results with supreme NMSE of 0.000697. Additionally, under the geographical area of 40sq.km, the Bi-LSTM scheme has attained least NMSE of 0.000190 and the LSTM scheme has obtained poor results with the higher NMSE of 0.000663.

TABLE I
TRAFFIC PREDICTION ANALYSIS OF EXISTING WITH PROPOSED BI-LSTM IN TERMS OF NMSE

Geo. Area (Sq. km)	LSTM	Bi-LSTM
5	0.013414	0.011046
10	0.006075	0.004350
15	0.004147	0.002659
20	0.003640	0.001847
25	0.001948	0.000799
30	0.000799	0.000257
35	0.000697	0.000224
40	0.000663	0.000190

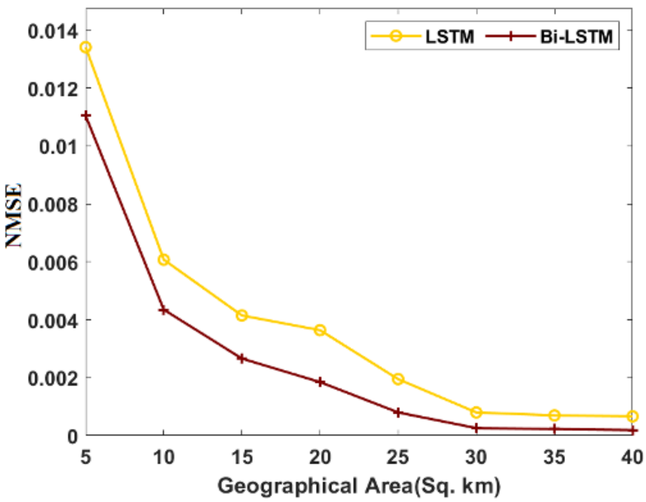

FIGURE 5. The NMSE analysis of Bi-LSTM with LSTM models.

For testing the data offloading performance of the ASCE method, an experimental analysis takes place interms of task completion under a varying number of tasks. Table 2 and Fig. 6 showcases the task completion time analysis of the ASCE model with other methods. The experimental values depicted that the ASCE model has required minimum task completion time over the other models. For instance, on the completion of 1 task, the ASCE model needs a lower completion time of 0.423560 whereas the adaptive and fixed models require a higher completion time of 0.675060 and 0.650721 respectively. Similarly, on the completion of 5 operations, the ASCE method requires the least completion time of 0.512802 while the adaptive and fixed methodologies need maximum completion time of 1.153722 and 1.251077 correspondingly. Followed by, on completion of 10 tasks, the ASCE approach demands minimal completion time of 0.626383 while the adaptive and fixed schemes need greater completion time of 1.559368 and 1.745965 respectively. Meantime, on completion of 15 tasks, the ASCE scheme necessities lesser completion time of 0.837318 and the adaptive as well as fixed techniques desires for a greater completion time of 2.175950 and 2.435563 correspondingly. Additionally, on the completion of 20 tasks, the ASCE approach requires minimal completion time of 0.983351 and the adaptive and fixed frameworks need greater completion time of 2.971015 and 3.198177 respectively.

TABLE II
RESULT ANALYSIS OF EXISTING WITH PROPOSED ASCE IN TERMS OF COMPLETION TIME OF TASK

Number of Tasks	Adaptive	Fixed	ASCE
1	0.675060	0.650721	0.423560
2	0.942786	0.845431	0.431673
3	0.975238	1.015803	0.456011
4	1.088819	1.096932	0.472237
5	1.153722	1.251077	0.512802
6	1.226738	1.283529	0.529028
7	1.291642	1.421448	0.561479
8	1.405222	1.567481	0.585818
9	1.397110	1.575594	0.610157
10	1.559368	1.745965	0.626383
11	1.567481	1.810868	0.658834
12	1.713513	1.932562	0.715625
13	2.046143	2.208401	0.804867
14	1.981240	2.200288	0.804867
15	2.175950	2.435563	0.837318
16	2.289530	2.508579	0.861657
17	2.403111	2.622160	0.869770
18	2.597821	2.833096	0.894109
19	2.897999	3.141387	0.950899
20	2.971015	3.198177	0.983351

Table 3 and Figs. 7-9 investigates the cyberattack detection performance of the proposed BMO-DBN model on the applied dataset. On determining classifier results interms of sensitivity, the DBN model has appeared as a poor classifier, which has obtained a minimum sensitivity of 91.45%. At the same time, the RF and RBF Network models have attained slightly better and closer sensitivity values of 92.39% and 93.40% respectively. Along with that, the RT and DT models have demonstrated moderate sensitivity values of 95.68% and 95.68%. Simultaneously, the LR model has exhibited manageable and competitive classification performance with a sensitivity of 97.26%. However, the presented BMO-DBN model has accomplished effective cyber-attack detection with a maximum sensitivity of 99.04%.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3030726, IEEE Access

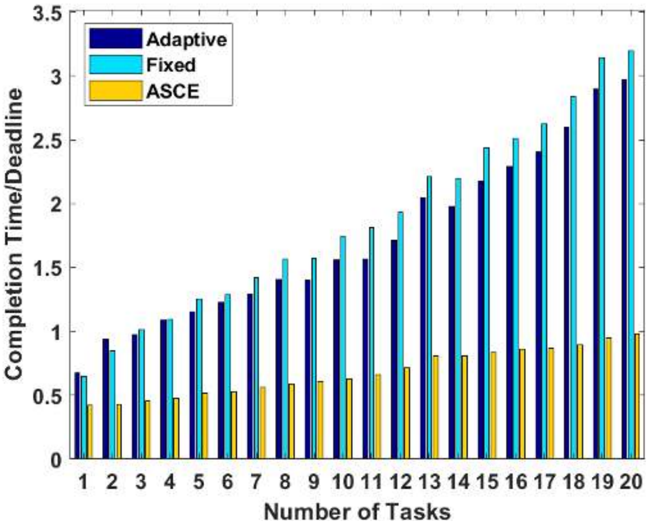

FIGURE 6. The Completion time analysis of ASCE model.

TABLE III
PERFORMANCE EVALUATION OF EXISTING WITH PROPOSED CLASSIFIERS ON ATTACK DETECTION DATASET

Classifier	Sensitivity	Specificity	Accuracy	F-score	Kappa
BMO-DBN	99.04	96.16	97.65	97.77	95.30
DBN	91.45	95.78	96.17	91.56	91.32
RBF Network	93.40	92.38	92.93	93.38	85.79
LR	97.26	96.92	97.10	97.29	94.19
RF	92.39	93.83	93.04	93.58	85.99
RT	95.68	95.39	95.55	95.84	91.06
DT	95.68	95.37	95.53	95.83	91.03

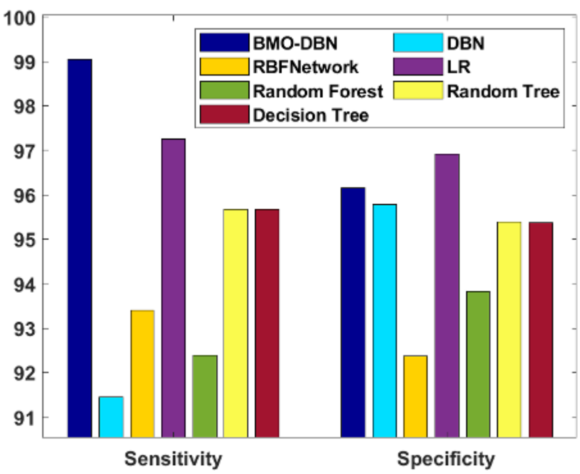

FIGURE 7. The Sensitivity and specificity analysis of BMO-DBN model.

On evaluating the simulation outcome with respect to specificity, the RBF Network method is referred to as a poor classifier, which has attained a lower specificity value of 92.38%. Simultaneously, the RF and DT methodologies have reached moderate and identical specificity measures of 93.83% and 95.37% correspondingly. In line with this, RT and DBN approaches have illustrated better specificity values of 95.39% and 95.78%. At the same time, the LR approach has represented a considerable and competing classification function with the specificity measure of 96.92%. Therefore, the projected BMO-DBN technology has gained optimal cyberattack prediction with higher specificity of 96.16%.

On computing classification outcomes by means of accuracy, the RBF Network approach is referred to as a worse classification model that has accomplished a lower accuracy of 92.17%. Meanwhile, the RF and DT schemes have gained considerable and nearby accuracy measures of 93.04% and 95.53% correspondingly. Likewise, the RT and DBN technologies have depicted better accuracies of 95.55% and 96.17%. Concurrently, the LR technique has represented acceptable as well as competing classification function with the accuracy value of 97.10%. Thus, the newly provided BMO-DBN scheme has achieved maximum cyberattack prediction with greater accuracy of 97.65%. On estimating classifier outcomes with respect to F-score, the DBN method is found to be inferior classification model that has attained lower F-score of 91.56%. Meanwhile, the RF and RBF Network methodologies have achieved moderate and similar F-score of 93.58% and 93.38% correspondingly. In line with this, the RT and DT technologies have depicted better F-scores of 95.84% and 95.83%. Meantime, the LR approach has represented acceptable and competitive classification function with the F-score of 97.29%. Hence, the projected BMO-DBN framework has attained standard cyberattack prediction with greater F-score of 97.77%.

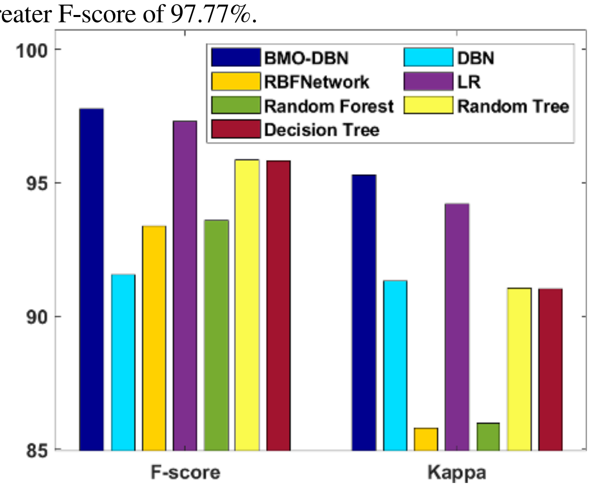

FIGURE 8. The F-score and kappa analysis of BMO-DBN model.

On calculating classification outcomes by means of kappa, the RBF Network scheme is said to be an ineffective classifier, which has accomplished lower kappa of 85.79%. Concurrently, the RF and DT technologies have gained better kappa values of 85.99% and 91.03% correspondingly. In line with this, the RT and DBN approaches have depicted moderate and identical kappa values of 91.06% and 91.32%. At the same time, the LR technique has depicted considerable and competing classification function with the kappa of 94.19%. Therefore, the projected BMO-DBN scheme has attained optimal cyberattack forecasting with the optimal kappa of 95.30%. The overall experimental results indicated the effective performance of the proposed model over the compared methods.

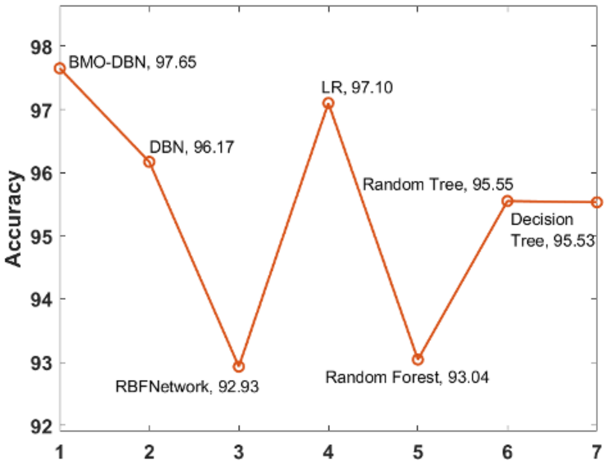

FIGURE 9. **The Accuracy analysis of BMO-DBN model.**

IV. CONCLUSION

This paper has developed a new DLTPDO-CD technique to predict traffic, offload data, and detect cyber-attacks in MEC. The presented model operates on three major phases. In the first stage, the traffic load prediction process is carried out using BiLSTM, which helps to achieve an effective data offloading process. Followed by, the ASCE model is applied for throughput maximizations for offloading users. At last, the DBN model is applied as the cyberattack detector and the learning rate of the DBN model is optimized by the BMO algorithm. Finally, the softmax classification layer is applied to perform the classification task. A series of experimentation takes place for verifying the goodness of the presented DLTPDO-CD model. The simulation outcome ensured the efficiency of the presented model and the cyber-attacks are detection with the sensitivity of 99.04%, specificity of 96.16%, accuracy of 97.65%, F-score of 97.77%, and kappa of 95.30%. In future, the presented model can be enhanced by the use of other traffic load prediction techniques.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their critical and constructive comments, their thoughtful suggestions have helped improve this paper substantially. Irina V. Pustokhina is thankful to the Department of Entrepreneurship and Logistics, Plekhanov Russian University of Economics, Moscow, Russia. Dr. K. Shankar would like to thank RUSA PHASE 2.0, Dept. of Edn. Govt. of India.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest. The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript.

REFERENCES

[1] A. Musaddiq, Y.B. Zikria, O. Hahm, H. Yu, A.K. Bashir, S.W. Kim, A Survey on resource management in IoT operating systems, IEEE Access 6 (2018) 8459–8482.

[2] A.A. Zaidan, B.B. Zaidan, M.Y. Qahtan, O.S. Albahri, A.S. Albahri, M. Alaa, F.M. Jumaah, M. Talal, K.L. Tan, W.L. Shir, C.K. Lim, A survey on communication components for IoT-based technologies in smart homes, Telecommun. Syst. 69 (July (1)) (2018) 1–25.

[3] P. Mach, Z. Becvar, Mobile edge computing: a survey on architecture and computation offloading, IEEE Commun. Surv. Tutor. 19 (3) (2017) 1628–1656.

[4] K. Sakthidasan, N. Vasudevan, P.K. GuruDiderot, C. Kadhiravan, "WOAPR: an affinity propagation basedclustering and optimal path selection for time-critical wireless sensor networks, IET Networks 8 (January (2)) (2019) 100–106.

[5] J. Mocnej, M. Miškuf, P. Papcun, I. Zolotová, Impact of Edge Computing Paradigm on Energy Consumption in IoT, IFAC-PapersOnLine 51 (6) (2018) 162–167.

[6] H. Zhang, J. Guo, L. Yang, X. Li, H. Ji, Computation offloading considering fronthaul and backhaul in small-cell networks integrated with MEC, in: Computer Communications Workshops, 2017, pp. 115–120.

[7] F. Wang, J. Xu, X. Wang, S. Cui, Joint offloading and computing optimization in wireless powered mobile-edge computing systems, IEEE Trans. Wireless Commun. 17 (2017) 1784–1797.

[8] H. Guo, J. Liu, H. Qin, X. Li, H. Ji, Collaborative computation offloading for mobile-edge computing over fiber-wireless networks, in: IEEE GLOBECOM, 2017.

[9] A. Khanna, A. Kero, D. Kumar, A. Agarwal, Adaptive mobile computation offloading for data stream applications, in: International Conference on Advances in Computing, communication and Automation, 2017, pp. 1–6.

[10] K. Zhang, Y. Mao, S. Leng, A. Vinel, Y. Zhang, Delay constrained offloading for Mobile Edge Computing in cloud-enabled vehicular networks, in: International Workshop on Resilient Networks Design and Modeling, 2016, pp. 288–294.

[11] X. Yang, Z. Chen, K. Li, Y. Sun, N. Liu, W. Xie, Y. Zhao, Communicationconstrained mobile edge computing systems for wireless virtual reality: scheduling and tradeoff, IEEE Access 6 (2018) 16665–16677.

[12] Y. Wu, Y. He, L.P. Qian, J. Huang, X. Shen, Optimal resource allocations for mobile data offloading via dual-connectivity, IEEE Trans. Mobile Comput. 17 (2018) 2349–2365.

[13] H. Izumikawa, J. Katto, RoCNet: Spatial mobile data offload with userbehavior prediction through delay tolerant networks, in: 2013 IEEE Wireless Communications and Networking Conference, WCNC, 2013, pp. 2196–2201.

[14] G. Feng, F. Xia, Y. Zhang, D. Su, H. Lv, H. Wang, H. Lv, Optimal cooperative wireless communication for mobile user data offloading, IEEE Access 6 (2018) 16224–16234.

[15] K. Lee, J. Lee, Y. Yi, I. Rhee, S. Chong, Mobile data offloading: how much can wifi deliver? IEEE/ACM Trans. Netw. 21 (2013) 536–550.

[16] Y. Zhang, F. Hou, L.X. Cai, J. Huang, QoS-based incentive mechanism for mobile data offloading, in: GLOBECOM 2017-2017 IEEE Global Communications Conference, 2017, pp. 1–6.

[17] H. Jang, C. Chang, Context aware mobile data offload using SDN, in: Telecommunication Networks and Applications Conference, 2017, pp. 185–190.

[18] X. Zhang, L. Guo, M. Li, Y. Fang, Motivating human-enabled mobile participation for data offloading, IEEE Trans. Mobile Comput. 17 (2017) 1624–1637.

[19] Siami-Namini, S., Tavakoli, N. and Namin, A.S., 2019, December. The performance of LSTM and BiLSTM in forecasting time series. In 2019 IEEE International Conference on Big Data (Big Data) (pp. 3285-3292). IEEE.

[20] Zhu, S., Xu, W., Fan, L., Wang, K. and Karagiannidis, G.K., 2019. A Novel Cross Entropy Approach for Offloading Learning in Mobile Edge Computing. IEEE Wireless Communications Letters, 9(3), pp.402-405.

[21] Yu, J. and Liu, G., 2020. Knowledge extraction and insertion to deep belief network for gearbox fault diagnosis. Knowledge-Based Systems, p.105883.

[22] Sulaiman, M.H., Mustaffa, Z., Saari, M.M. and Daniyal, H., 2020. Barnacles Mating Optimizer: A new bio-inspired algorithm for solving engineering optimization problems. Engineering Applications of Artificial Intelligence, 87, p.103330.