Research Article

Hybrid Metaheuristic Optimization based Feature Subset Selection with Classification Model for Intrusion Detection in Big Data Environment

[1]B.Vijaya Kumar, [2]Dr. S. Mohan

[1]Research Scholar, Department of Computer Science and Engineering, Annamalai University Tamilnadu, India.
[2]Assistant Professor, Department of Computer Science and Engineering, Annamalai University Tamilnadu, India.

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 23 May 2021

Abstract

Big Data denotes an enormous set of distinct structured data attained from various heterogeneous sources stacked on the storage devices ranging from petabytes to zetabytes. Employing security in Big Data is highly a crucial process due to an exponential increase in data sizes. Therefore, intrusion detection system (IDS) is employed to detect the presence of intrusions on computers, workstations, or networks. In this view, this paper presents a Hybrid Metaheuristic Optimization based Feature Subset Selection (HMOFS) with an Optimal Wavelet Kernel Extreme Learning Machine (OWKELM) based Classification model called HMOFS-OWKELM model for IDS in big data environment. In order to handle big data, Hadoop Ecosystem is utilized. The proposed HMOFS-OWKELM model involves preprocessing to remove the unwanted noise that exists in it. In addition, the HMOFS includes the hybridization of moth flame optimization (MFO) with hill climbing (HC) based feature selection process. The HC concept is incorporated to the MFO algorithm to enhance the convergence rate. Besides, OWEKM model is applied for classification process where the optimal parameter setting in the WKELM is carried out by the rat swarm optimizer (RSO). A wide range of simulations was performed on the benchmark NSL-KDDCup dataset and the results are examined interms of different evaluation parameters. The obtained results showcased that the HMOFS-OWKELM model outperforms the other methods by offering a maximum detection accuracy of 99.67%.

1. Introduction

Big Data is information which is complex to save, control, and investigate by conventional software and database methods. It involves velocity and high volume, and collection of information which requires novel method to handle it. An intrusion Detection System (IDS) is software or hardware tool which investigates the information to identify the attackers over network or system. Conventional IDS methods become difficult and ineffective while handling Big Data since its investigation procedure is difficult and lengthy [1, 2]. Big Data methods and approaches to investigate and save information in IDS could decrease training and computational time. The IDS comprises 3 techniques for detecting attacks namely Hybrid, Anomaly, and Signature based detection. The signature based detection is implemented to identify acknowledged attacks utilizing signature of individual’s attack. It is an efficient technique for identifying the acknowledged attack which is pre-loaded in IDS database. Thus, it is always assumed as highly precise in detecting the intrusion effort of acknowledged attacks [3]. But this technique does not detect the new attacks as its signatures cannot exist and the databases are often upgraded to raise the efficiency of identification [4].

To conquer this issue, Anomaly-based identification technique links the present client actions towards predetermined profile which is utilized to identify abnormal behavior of the intrusions. The technique is efficient towards anonymous or zero-day attacks with no other upgrades to the method. But this technique generally has high false positive rate (FPR) [5, 6]. Hybrid-based identification is an integration of several techniques of IDS to conquer the drawback in the individual system utilized and attain the benefits of numerous techniques. Various works have presented Machine Learning (ML) techniques to decrease FPR and generate precise IDS. But, to handle Big Data, the ML conventional methods are time consuming in learning and categorizing information. For resolving several problems in big data like speed and computation time, accurate IDS needs to be developed. The goal of this technique is to present Spark Big Data methods which handle Big Data in IDS to decrease computational time and attain efficient classifiers.

The security attack is initiated nearly on daily basis and performed in many practices such as spyware, worms, virus, ransomware, rootkits, Structured Query Language (SQL) injection, Denial of Service (DoS), cross site scripting, buffer overflow, masquerading, etc. The non-determined behavior attacks have defined several complexes to identify and respond. Several methods and systems are implemented to solve these privacy problems involving IDS, firewall, packet sniffers, antivirus, anti-rootkits, and encryption methods [7]. In this method, IDS choose another cyber security methods for several purposes amongst topnotch which covers entire features of a network traffic and Internet Protocol (IP) packet and which need to differentiate among genuine and malicious packets involving corresponding of pre saved signature attack which decreases the false alarm rate to be reasonable [8, 9]. The IDS are often being the main consideration and critical concept of study relating to cyber-attack over computers, workstations, and networks. It consists of 2 kinds namely Signature and Anomaly based IDS.

Scientists have processing towards all Hybrid-IDS to improve the efficiency with respect to false alarm, detection rate, and accuracy which assist in building additional strength and trustworthy method. This presented method, it training the classification by ML to verify the string, regular expression, and digital signature when Signature based IDS and authorizing the traffics of received packet for Anomaly based IDS [10]. In order to attain a successive reinforced IDS, it utilizes automatic classification in Apache Spark. It utilizes NSL KDD cup 99 datasets [11] are the derivative of KDD cup 99 datasets [12]. It similarly preprocesses to comfort, for instance with respect to altering string value into mathematical value for convenient performance of feature selection process. The FS utilizes NSGA-II [13] that is Pareto optimum solution based method. Multi-modal fusion [14] is the procedure for combining several input models to integrate into a whole instruction. The combination method is depending upon Artificial Neural Networks (ANN), statistical method, Hidden Markov technique, time-stamped lattice, or finite state transducer, based on methods. It is generally consisting of 3 methods to perform multi-modal fusion involving recognition, decision, and hybrid multi-level fusion.

This paper presents a Hybrid Metaheuristic Optimization based Feature Subset Selection (HMOFS) with an Optimal Wavelet Kernel Extreme Learning Machine (OWKELM) based Classification model called HMOFS-OWKELM model for IDS in big data environment. In order to handle big data, Hadoop Ecosystem is utilized. The proposed HMOFS-OWKELM model involves preprocessing to remove the unwanted noise that exists in it. In addition, the HMOFS includes the hybridization of moth flame optimization (MFO) with hill climbing (HC) based feature selection process. Besides, OWEKM model is applied for classification process where the optimal parameter setting in the WKELM is carried out by the rat swarm optimizer (RSO). A wide range of simulations was performed on the benchmark NSL-KDDCup dataset.

2. The Proposed HMOFS-OWKELM model

Fig. 1 illustrates the workflow involved in the HMOFS-OWKELM model. The figure states that the input networking data is preprocessed to discard the existence of noise exist in it. Followed by, the HMOFS algorithm gets executed to select an optimal set of features by incorporating the characteristics of MFO and HC algorithms. Once the feature subsets were chosen, they are fed into the WKELM model to determine the existence of intrusions in the network. At last, RSO is employed to enhance the detection efficiency by determining the optimal hyperparameter values of the WKELM model

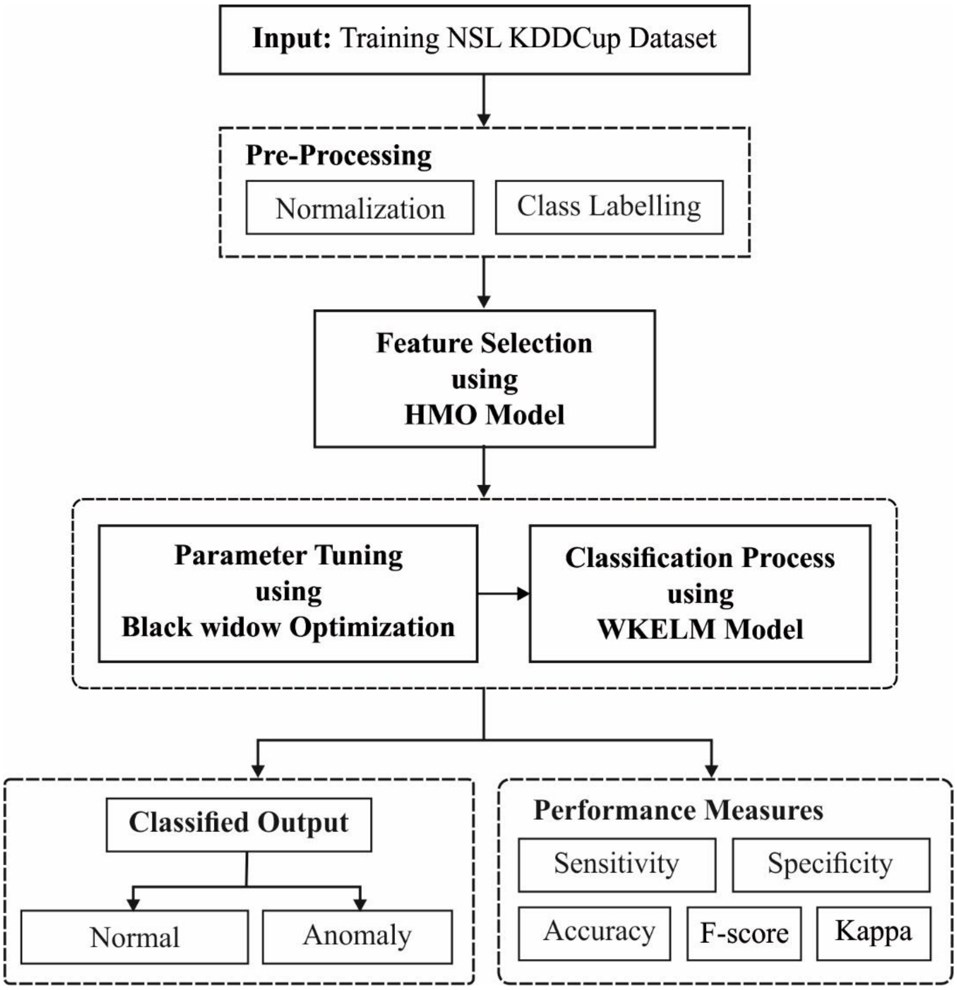
Input: Training NSL KDDCup Dataset
Pre-Processing
Normalization
Class Labelling
Feature Selection using HMO Model
Parameter Tuning using Black widow Optimization
Classification Process using WKELM Model
Classified Output
Normal
Anomaly
Performance Measures
Sensitivity
Specificity
Accuracy
F-score
Kappa

Fig. 1. Steps in Proposed Model

2.1. Hadoop Ecosystem

In order to manage Big Data, Hadoop Ecosystem and its components are extremely utilized. During the distributed background, Hadoop is the type of open-source structure which allows stakeholders for storing and processing the Big Data on computer clusters by utilizing simple programming methods. To thousands of nodes in single server, it can be exhibited to contain improved scalability and fault tolerance. 3 important elements of Hadoop are MapReduce, Hadoop Distributed File System (HDFS), and Hadoop YARN [22-26].

2.2. HMOFS based Feature Subset Selection

The preprocessed networking data is provided as input to the HMOFS model, which incorporates the characteristics of MFO and HC algorithms.

2.2.1. Hill Climbing

To speed up the convergence rate, the moth swaps with prior optimum location using local minimum of its present location. A local minimum is attained by HC. The location is captured as a renumbering of vertices in the context of graph. The HC process would swap the label of 2 vertices when it identifies the bandwidth and it will decrease accordingly. They observe major features of the minimization problem which might be several labeling f with similar $B_{f(G)}$ value; especially there might be many vertices v including $B_{f(v)} = B_{f(G)}$; they called this vertex as critical vertex. A swapping decreases the matrix bandwidth if it decreases the bandwidth of critical vertices. Hence, the HC process initially identifies the critical vertices, later seeks appropriate vertices for swap.

An appropriate swap vertex list for critical vertex v to label f can be defined by

$$\begin{aligned}\max(v) &= \max\{f(u) : u \in N(v)\}\\ \min(v) &= \min\{f(u) : u \in N(v)\}\end{aligned} \tag{1}$$

As the optimal label of v in present label f is

$$mid(v) = \left\lfloor \frac{(\max(v) + \min(v))}{2} \right\rfloor, \tag{2}$$

Later the group of appropriate swap vertices for v is given as

$$N'(v) = \{u : |mid(v) - f(u)| < |mid(v) - f(v)|\} \tag{3}$$

It is considered that every vertex u with label $f(u)$ which is nearer to $mid(v)$ than $f(v)$. In HC, they arrange $N'(v)$ in ascending order based on $|mid(v) - f(u)|$ value, later tries to interchange $f(v)$ with $f(u)$ individually in the arranged order. When a swapping is adopted, it will end with the following critical vertex. A swapping label of v and u is adopted afterward the swapping, v turn into non-critical, u is not altered to a critical vertex (if it is critical beforehand the swapping), and the bandwidth isn't raised. The time complexity for every iterations of HC method is $O(|V|^2)$ in the worst. As there are $|V|$ vertices, the bandwidth could be decreased in all $|V|$ times, and vertices $|V|$ is decreased from critical to non-critical. Thus, iterations of HC method are bounded by $O(|V|^2)$. Then the complexity of HC method is $O(|V|^4)$ in the worst. Indeed, they operate on sparse graph/matrix, the HC method runs more quickly compared to worst case complexity. In huge conditions, they could fix a constraint for the iterations to raise the speediness of HC.

2.2.2. MFO algorithm

The MFO is the latest metaheuristic technique that imitates the navigation system of moth in nature. The MFO begins with a population of N moth, and every moth in the population is assumed as possible result $(x_i, i = 1, \dots, N)$ with Dim decision parameters x_{ij}, $(j = 1, \dots, Dim)$. Based on [15], the search technique is scientifically defined as 3 tuples:

$$MFO = (R, B, T), \tag{4}$$

Where R denotes initial stage, in which the population of moths is created arbitrarily, and FF for every moth is calculated. Later, the better location of moth is stored in flames. The B denotes upgrading stage, in which moth is upgraded by the movement near the search domain, whereas T utilized to denote stopping condition. Additionally, B stage, the location of moth is upgraded according to the flames in Eq. (5):

$$x_i = D_i e^{bl} \cos(2\pi l) + F_u, i, u = 1, \ldots, N \tag{5}$$

$$D_i = |F_u - x_i|, \tag{6}$$

Where $l \in [-1, 1]$ and b denotes uth fame, an arbitrary number and constant is utilized to describe the shape of the logarithm spiral function, correspondingly. Since fame counts ($Flames_N$) might be reduced to resolve the issue of degrading the utilization of optimum position, and issue happened because of upgrading $x_i,\ i = 1, \ldots, N$ which is carried out with respect to distinct N positions. The adaptive scheme is utilized to overcome the issue as determined in [16]:

$$Flames_N = round\left(N - C \times \frac{N-1}{C_{\max}}\right) \tag{7}$$

Where C denotes current iteration and $C_{\max}$ represents highest iteration count (i.e. T phase).

2.2.3. Application of HMOFS for Feature Selection

An alternate technique to deal the FS challenge is presented MFO technique and DE technique as the approach for localizing search for MFO method. Because of this, the MFO examines the search space more than utilizing it; therefore the integration enhances the performance of MFO and prevents it from being trapped in local points. This technique comprises of initial stage, upgrading stage, and classification stage. Those stages of presented method are provided with additional detail in next subdivisions.

Initial phase

Generally, the early population is significant at all MH techniques as it affects convergence rate and the quality of end result. Here, the MFO begins with generating a population X, with N solution and dimensions of all solutions is Dim:

$$X = L + (U - L) \odot rand(N, Dim), \tag{8}$$

Where U represent maximum limit and L indicates minimum limit of searching area. To fulfill this phase, the presented approach should transform the continuous values of solution to binary solution, this is carried out by the next equation:

$$x_i = \begin{cases} 1\ if\ \dfrac{1}{1+e^{-x_i}} > \sigma \\ 0\ otherwise, \end{cases} \tag{9}$$

Where σ denotes threshold value (here, it is set as 0.55,). For instance, it assumed the value of x_i are [0.93, 0.33, 0.61, 0.54, 0.28, 0.34, 0.86], later the output of Eq. (9) denotes$[1, 0, 1, 0, 0, 0, 1]$. It implies that the feature in dataset corresponds to selected ones as related features and respective to 0’s are neglected.

Afterward FS, the FF is employed to calculate the efficiency of this feature, the FF utilized in presented technique is described in Eq. (10):

$$f(x_i) = \xi \times Err_{x_i} + (1 - \xi) \times \left(\frac{|x_i|}{Dim}\right) \tag{10}$$

Where Err_{x_i} represents classifier error, $|x_i|$ describes the FS count and Dim denotes overall feature count. The ξ represents arbitrary value lies in [0, 1], which is utilized for balancing the classification accurateness and FS count.

The initial phase is calculating the possibility of objective function values ($Prob_i, i = 1,2, \ldots, N$) for every component in the population as:

$$Prob_i = \frac{f_i}{\sum_{i=1}^{N} f_i} \tag{11}$$

The following phase in upgrading stage is to upgrade the present result based on the possibility value $Prob_i$. For instance, when $Prob_i$ value is larger when compared to δ, after that the MFO is utilized; or else, the HC is utilized. This implies the quality of present outcome is lower when compared to δ, thus the operator of HC technique is utilized to enhance its utilization capability. The produced vector is distributed to FF (Eq. (10)) to define its quality for selecting the optimum result x_{best}. Therefore, this series of phase, in upgrading stage, is iterated till it reaches the highest iteration count that is assumed as stopping criterion of the operator. At the classification stage, the FS (i.e. optimum result x_{best}) are fed into the classification model, which is discussed in the subsequent sections.

2.3. WKELM based Classification

Once the feature subsets were chosen, they are fed into the WKELM model to determine the existence of intrusions in the network. Huang [17] presented a feed-forward NN named ELM, where output weight can be adapted only at the time of training that is arbitrarily allocated radial basis function with hidden neurons, threshold voltages, and weights. The ELM is a common single hidden layer feed forward networks (SLFN) method and hidden layer is unnecessary to determine.

For N arbitrarily distinct instances(x_i, t_i), standardized by H hidden node and activation function $h(x)$ represents: $x_i = \left[x_{i1}, x_{i2}, \dots x_{jD}\right]^T \in R^D$ and $f_i = [t_{i1}, \mathrm{t}_{i2}, \dots, \mathrm{t}_{\mathrm{ik}}]^T \in R^K$. The SLFNs is shown in Eq. (12):

$$\sum_{i=1}^{L} \beta_i\, h_i(x_j) = \sum_{i=1}^{L} \beta_i\, h_i(w_i x_j + b_i) = o_j, j \in 1, N \tag{12}$$

In contrast to the principle of NN, every hidden node is needed to be fixed in the FNNN displays the hidden node/network that is arbitrarily made in ELM learning concept. Each hidden node variable is not dependent on the targeted function or trained dataset. The hidden node/neuron variables in ELM are independent in nature whereas the standard FFNN with hidden nodes contains extensive method and partition ability. The KELM architecture is shown in Fig. 2.

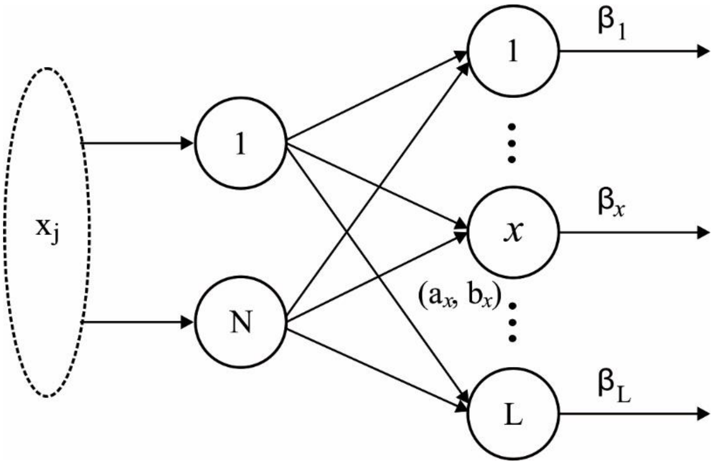

Fig. 2. Structure of KELM

In Eq. (12), $w_i = [w_{i1}, \mathrm{w}_{i2}, \dots \mathrm{w}_{iD}]^T$, is the weight vectors linked to the hidden as well as input nodes, $\beta_i = [\beta_{i1}, \beta_{i2}, \cdots \mathrm{w}_{iK}]^K$ represents weight vector linked to ith and output node, and b_i represents the ith node. The SLFN with activation function $h(x)$ and hidden node L is efficiently defined in Eqs. (13)-(15).

$$H\beta = T \tag{13}$$

$$H = \begin{bmatrix} h_1(w_1 x_1 + b_1) & \cdots & h_1(\mathrm{Lx}_1 + b_\mathrm{L}) \\ \vdots & \ddots & \vdots \\ h_1(w_1 x_N + b_1) & \cdots & h_L(w_L x_N + b_L) \end{bmatrix}_{N \times M} \tag{14}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{M \times 1} \quad AND\ T = \begin{bmatrix} t_1^T \\ \vdots \\ t_L^T \end{bmatrix}_{N \times 1} \tag{15}$$

Additionally, ELM has resolved the function in Eq. (16),

$$\beta = H \dagger T \tag{16}$$

where, $H^\dagger$ denotes Moore Penrose, the general inverse of H matrix to attain minimum square outcome.

The kernel approach, the latest method for ELM, is in great need and applied to a variety of methods. According to investigation carried out by extending ELM to kernel learning, the ELM uses several feature mappings involving arbitrary hidden nodes and kernels. On comparing with the ELM, the KELM is capable of maintaining a definite mapping, by kernel functions, rather than assuming a certain mapping connection [18]. The training error and output weight is decreased via kernel, whereas the entire effectiveness of NN could be increased as displayed in Eq. (17);

$$Min: ||H\beta - T||, ||\beta|| \tag{17}$$

Eq. (17); is systematized by least square outcomes in Eq. (18);

$$\beta = H^T \left[\frac{1}{c} + HH^T\right]^{-1} T \tag{18}$$

where C denotes adaptation coefficients, T represents outcome matrix, and H indicates hidden layered output matrix. The functional outcome of the ELM is expressed as

$$f(x) = h(x)H^T \left[\frac{1}{c} + HH^T\right]^{-1} T \tag{19}$$

If h(x) cannot be identified in feature mapping, the kernel matrices of ELM is explained based on Mercer's condition as denoted in Eq. (20);

$$M = HH^T: m_{ij} = H(x_j) = \mathrm{k}(x_i, x_j) \tag{20}$$

So, the output function $f(x)$ is expressed as;

$$f(x) = [k(x, x_1) \dots k(x, x_N)] \left[\frac{1}{c} + M\right]^{-1} T \tag{21}$$

In ELM, a majority of kernel based functions like exponential, linear, wavelet, and Gauss are acquired from Mercer situation. $M = HH^T$ and $k(\mathrm{x}, \mathrm{y})$ denotes kernel functions of SLFN. The input data is categorized by wavelet function KLEM. The KLEM is displayed in Eq. (22);

$$k(\mathrm{x}, \mathrm{y}) = \cos\left[\alpha(||x - y||/\beta)\right] \exp\left[-(||x - y||^2/\theta)\right] \tag{22}$$

The variables perform a major function in enhancing the efficiency of the classifications, that is, β, and θ utilized in wavelet KELM technique [19].

2.4. Hyperparameter Optimization

For determining the optimal hyperparameter values of the WKELM model, RSO is employed to enhance the detection efficiency. Rats are medium-sized and long-tailed rodents that are distinct based on weight and size. Brown and Black rats are the two major classes. In rat's family, the female rats are called as does whereas male rats so-called bucks. They are commonly social intelligent in nature. Mutually, they include several actions like boxing, tumbling, jumping, and chasing. They live together in females and males' rats as territory animals. The performance of rats is extremely destructive in several events that might result in a decrease of few animals. These destructive actions are the major inspiration of this work whereas fighting and chasing by target. In this study, the fighting and chasing actions of rats are precisely demonstrated to achieve optimization and planning of RSO technique.

Commonly, they are societal animal that hunts the target in a set by their social agonistic actions. To describe their actions precisely, consider the optimum search factor which has the knowledge of position of the target. Another search agent upgrades their locations in terms of optimum search agent which is attained previously. The succeeding equations are presented by:

$$\vec{P} = \mathrm{A} \cdot \vec{\mathrm{P}}_i(\mathrm{x}) + \mathrm{C} \cdot \left(\vec{\mathrm{P}}_r(\mathrm{x}) - \vec{\mathrm{P}}_i(\mathrm{x})\right) \tag{23}$$

where $\vec{\mathrm{P}}_i(\mathrm{x})$ denote the positions of rats and $\vec{\mathrm{P}}_r(\mathrm{x})$ indicates optimum result. But, A and C variables are estimated by:

$$A = R - x \times \left(\frac{R}{\mathrm{Max}_{Iteration}}\right) \tag{24}$$

were, $x = 0, 1, 2, \cdots \mathrm{Max}_{Iteration}$

$$C = 2 \cdot rand() \tag{25}$$

Thus, C and R are arbitrary numbers which exist among [0, 2], and [1, 5] correspondingly. The variables of A and C are the responsibilities for best exploitation and exploration through the sequence of iteration. It precisely determines the aggressive course of rats by target and is presented by:

$$\vec{P}_i(x+1) = \left|\vec{P}_r(x) - \vec{P}\right| \tag{26}$$

where $\vec{P}_i(x+1)$ denotes the upgraded succeeding location of rat. It stores the optimum result and upgrades the locations of another search agent in terms of optimum search agents. Thus, exploitation and exploration are assured with the adapted value of variables A and C. The presented RSO technique stores the best result by some functions. The steps involved in the RSO are discussed as follows.

Step 1. Initial rat's population P_i where $i = 1, 2, \ldots, n$.
Step 2. Select the initialized variables of RSO algorithm: A, C, and R.
Step 3. Here, estimates the fitness value of every search agent.
Step 4. An optimum search agent is afterward explored in the provided search space.
Step 5. Upgrade the locations of search agent utilizing Eq. (26).
Step 6. To verify if there some search agents come over edge constraint of search space and after adapt it.
Step 7. Return, estimate the upgraded search agent fitness value, and upgrade the vector P_r whether it is an optimum solution over preceding better solutions.
Step 8. Stop the algorithm when the termination conditions are fulfilled. Or, return to Step 5.
Step 9. Return the optimally attained better solution.

3. Performance Validation

The experimental validation of the presented model takes place using the NSL-KDD dataset. It comprises a set of normal and anomaly instances. The details related to the dataset are given in Table 1.

Table 1 Types of Attacks in NSL-KDD Dataset

Attack Type	**Description**	**No. of Samples**
Anomaly Instances		
Dos	Denial of service attack	45,927
R2l	Unauthorized access from a remote host	995
Probe	Port monitoring or scanning	11,656
U2r	Unauthorized local super user privileged access	52
Normal Instances		
Normal	Not an Attack	67,343

Table 2 and Fig. 3 summarizes the result analysis of the HMO-FS model with other FS models on the test IDS dataset. From the results, it is obvious that the GA-FS model has resulted in worse outcomes by offering the maximum least best cost of 0.00115060. Concurrently, the BSO-FS and WOA-FS models have exhibited certainly improved performance with the closer best cost of 0.00079351 and 0.00093985 correspondingly. Finally, the HMO-FS model has demonstrated effective FS outcome by offering the least best cost of 0.00076323.

Table 2 Results of Existing with Proposed HMO-FS Method on Applied IDS Dataset

Methods	**Best Cost**	**Selected Features**
HMO-FS	0.00065210	2,4,5,6,7,9,,12,14,15,17,22,27,38
QBSO-FS	0.00076323	2,3,5,6,7,8,9,11,14,16,18,32,36,39

BSO-FS	0.00079351	2,4,5,6,7,9,11,13,15,16,17,19,20,21,23,38,38,40
WOA-FS	0.00093985	3,5,8,13,18,20,21,22,23,25,26,28,30,32,33,34,36,38,40
GA-FS	0.00115060	21,7,27,32,25,34,1,2,35,3,24,40,28,26,10,5,33,14,16,12,36,23,30,38,22,15,37,9

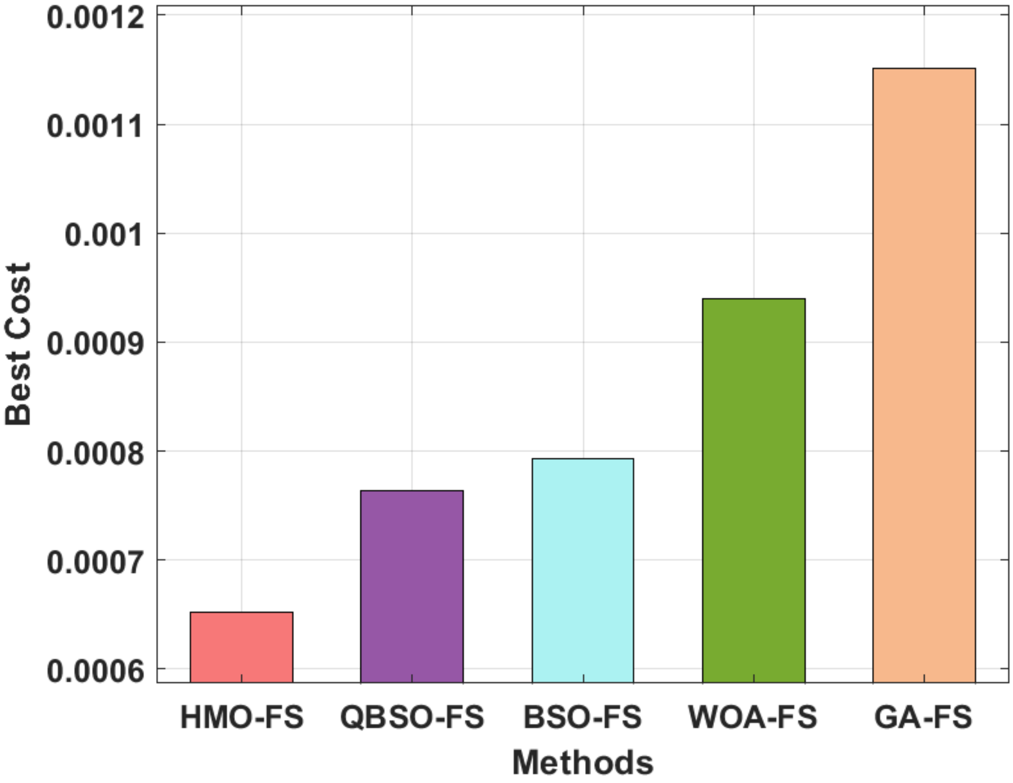

Fig. 3. Best cost analysis of HMOFS-OWKELM model

Table 3 and Fig. 4 investigates the IDS performance analysis of the HMOFS-OWKELM model interms of different measures. From the table, it can be clear that the HMOFS-OWKELM model has detected different types of intrusions effectively. For instance, the HMOFS-OWKELM model has identified the DoS attacks with the sens. of 99.35%, spec. of 99.32%, acc. of 99.33%, F-measure of 99.23%, and kappa of 98.46%. Besides, the HMOFS-OWKELM model has identified the DoS attacks with the sens. of 99.12%, spec. of 99.73%, acc. of 99.54%, F-measure of 99.31%, and kappa of 98.81%. Moreover, the HMOFS-OWKELM model has identified the R2l attacks with a sens. of 98.99%, spec. of 99.82%, acc. of 99.86%, F-measure of 99.46%, and kappa of 99.36%. Furthermore, the HMOFS-OWKELM model has identified the Probe attacks with the sens. of 99.89%, spec. of 99.82%, acc. of 99.86%, F-measure of 99.46%, and kappa of 99.36%. Eventually, the HMOFS-OWKELM model has identified the U2r attacks with the sens. of 98.89%, spec. of 99.91%, acc. of 99.90%, F-measure of 99.41%, and kappa of 99.45%. Overall, the HMOFS-OWKELM model has identified the attacks effectively by obtaining an average sens. of 99.18%, spec. of 99.71%, acc. of 99.67%, F-measure of 99.41%, and kappa of 99.14%.

Table 3 Result Analysis of Proposed HMOFS-OWKELM Model

Attacks	**Sensitivity**	**Specificity**	**Accuracy**	**F-Measure**	**Kappa**
Dos	99.35	99.32	99.33	99.23	98.46
R2l	99.12	99.73	99.54	99.31	98.81
Probe	98.99	99.82	99.86	99.46	99.36
U2r	98.89	99.91	99.90	99.41	99.45

Normal	99.56	99.76	99.72	99.64	99.62
Average	**99.18**	**99.71**	**99.67**	**99.41**	**99.14**

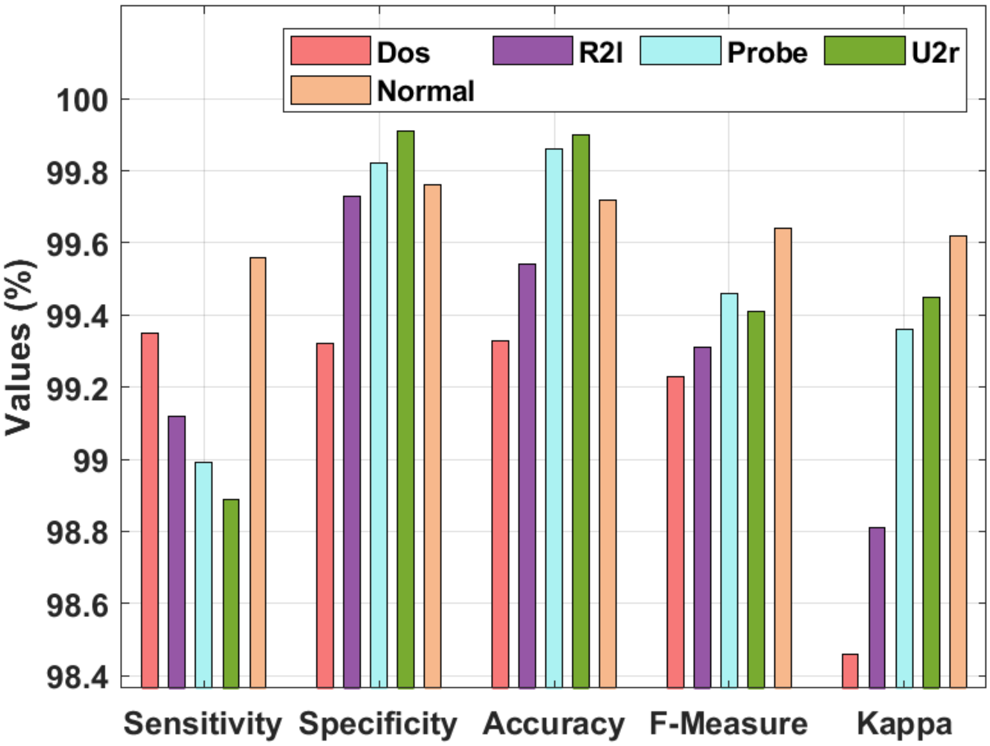

Fig. 4. Result analysis of HMOFS-OWKELM model

Table 4 and Fig. 5 give a detailed comparative outcomes analysis of HMOFS-OWKELM method with other existing techniques. On examining the detection outcomes with respect to accuracy, the RBFNetwork model has resulted in worst detection performance with an accuracy of 92.93% while a somewhat higher accuracy of 93.04% has been achieved by the Rand Forest approach. Followed by, the DT(J48) manner has demonstrated certainly superior accuracy of 95.53%. Concurrently, the Rand. Tree model has resulted in a manageable accuracy of 95.55%. But, the LR methodology has portrayed near optimal accuracy of 97.10%, the proposed HMOFS-OWKELM technique has reported a higher accuracy of 99.67%. On investigative the detection outcomes interms of sensitivity and specificity, the Rand. Forest technique has resulted in minimum detection performance with the sensitivity and specificity of 92.39% and 93.83% whereas a somewhat increased sensitivity and specificity of 93.4% and 92.38% has been achieved by the RBFNetwork model. In line with, the DT(J48) method has depicted certainly maximum sensitivity and specificity of 95.68% and 95.37%. Followed by, the Rand. Tree model has resulted in a manageable sensitivity and specificity of 95.68% and 95.39%. Also, the LR manner has showcased near optimal sensitivity and specificity of 97.26% and 96.92%, the presented HMOFS-OWKELM approach has reported a maximum sensitivity and specificity of 99.18% and 99.71%.

Table 4 Performance Analysis of Traditional Classifiers with Proposed HMOFS-OWKELM Model for Applied Dataset

Methods	**Sensitivity**	**Specificity**	**Accuracy**	**F-measure**	**Kappa**
Proposed HMOFS-OWKELM	99.18	99.71	99.67	99.41	99.14
RBFNetwork	93.40	92.38	92.93	93.38	85.79
LR	97.26	96.92	97.10	97.29	94.19
Rand. Forest	92.39	93.83	93.04	93.58	85.99

Rand. Tree	95.68	95.39	95.55	95.84	91.06
DT (J48)	95.68	95.37	95.53	95.83	91.03

On examining the detection results with respect to F-measure and kappa, the RBFNetwork model has resulted in least detection shown with the F-measure and kappa of 93.38% and 85.79% whereas a somewhat superior F-measure and kappa of 93.58% and 85.99% has been attained by the Rand. Forest model. Similarly, the DT(J48) approach has demonstrated certainly a maximal F-measure and kappa of 95.83% and 91.03%. On continuing with, the Rand. Tree technique has resulted in a manageable F-measure and kappa of 95.84% and 91.06%. Moreover, the LR approach has showcased near optimal F-measure and kappa of 97.29% and 94.19%, the presented HMOFS-OWKELM model has reported a higher F-measure and kappa of 99.41% and 99.14%.

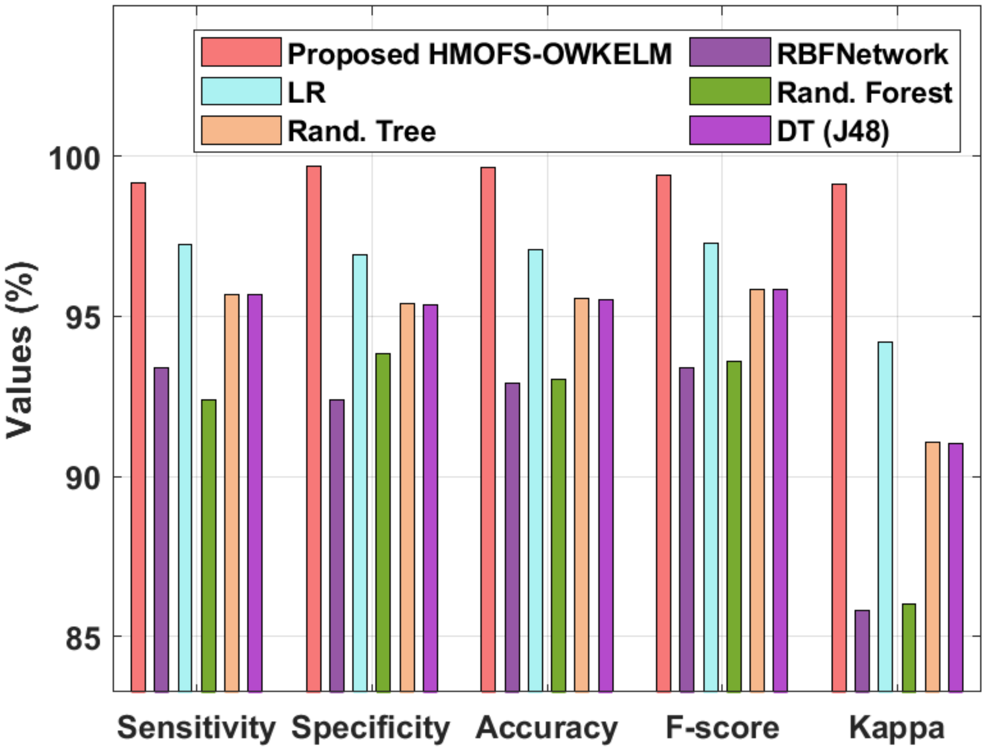

Fig. 5. Comparative analysis of HMOFS-OWKELM model with different measures

Table 5 and Fig. 6 demonstrate the comparison study of the HMOFS-OWKELM model with recent state of art techniques with respect to accuracy [20, 21]. From the obtained results, it is obvious that the CS-PSO and GBT techniques have established unimportant detection performance with the accuracy of 75.51% and 84.25% correspondingly. Additionally, the Gaussian Process and DNN-SVM techniques have represented somewhat better and nearer accuracy of 91.06% and 92.03% correspondingly. In the same way, the Fuzzy c-means, GA-Fuzzy, and Cuckoo optimization methods have ensured improvised results with the accuracy of 95.3%, 96.53%, and 96.88% correspondingly. Though the IntruDTree and Behaviour based IDS techniques have stated the near optimal accuracy of 98% and 98.80%, the presented HMOFS-OWKELM technique has demonstrated superior performance with the higher accuracy of 99.67%.

Table 5 Performance Analysis of Recent Methods with Proposed HMOFS-OWKELM Model on Applied Dataset

Methods	**Accuracy (%)**
Proposed HMOFS-OWKELM	99.67
IntruDTree (2020)	98.00

CS-PSO (2019)	75.51
Behaviour Based IDS (2019)	98.80
Cuckoo Optimization (2018)	96.88
DNN+SVM (2018)	92.03
GA+ Fuzzy (2018)	96.53
Fuzzy C-Means (2018)	95.30
Gradient Boosting (2018)	84.25
Gaussian Process (2015)	91.06

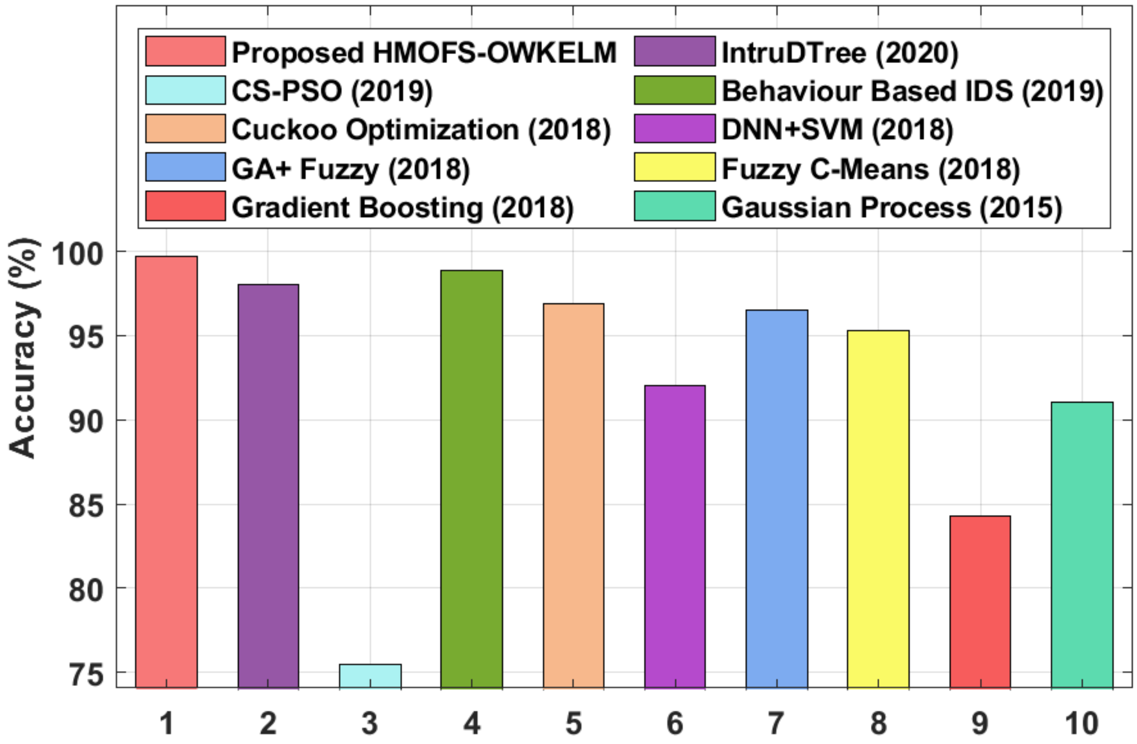

Fig. 6. Accuracy analysis of HMOFS-OWKELM model with recent techniques

4. Conclusion

This paper has presented an effective HMOFS-OWKELM model for intrusion detection in big data environment. Firstly, the input networking data is preprocessed to discard the existence of noise exist in it. Followed by, the HMOFS algorithm gets executed to select an optimal set of features by incorporating the characteristics of MFO and HC algorithms. Once the feature subsets were chosen, they are fed into the WKELM model to determine the existence of intrusions in the network. At last, RSO is employed to enhance the detection efficiency by determining the optimal hyperparameter values of the WKELM model. A wide range of simulations was performed on the benchmark NSL-KDDCup dataset and the results are examined interms of different evaluation parameters. The obtained results showcased that the HMOFS-OWKELM model outperforms the other methods by offering a maximum detection accuracy of 99.67%.

References

[1] Tchakoucht TA, Ezziyyani M. Building a fast intrusion detection system for high-speed-networks: probe and DoS attacks detection. Procedia Comput Sci. 2018;127:521–30.

[2] Zuech R, Khoshgoftaar TM, Wald R. Intrusion detection and big heterogeneous data: a survey. J Big Data. 2015;2:3.

[3] Sahasrabuddhe A, et al. Survey on intrusion detection system using data mining techniques. Int Res J Eng Technol. 2017;4(5):1780–4.

[4] Dali L, et al. A survey of intrusion detection system. In: 2nd world symposium on web applications and networking (WSWAN). Piscataway: IEEE; 2015. p. 1–6.

[5] Scarfone K, Mell P. Guide to intrusion detection and prevention systems (idps). NIST Spec Publ. 2007;2007(800):94.

[6] Debar H. An introduction to intrusion-detection systems. In: Proceedings of Connect, 2000. 2000.

[7] Suthaharan S. Big data classification: Problems and challenges in network intrusion prediction with machine learning. ACM SIGMETRICS Perform Eval Rev 2014;41(4):70–3.

[8] Zuech R, Khoshgoftaar TM, Wald R. Intrusion detection and big heterogeneous data: a survey. J. Big Data 2015;2(1):3.

[9] Jonnalagadda SK, Reddy RP. A literature survey and comprehensive study of intrusion detection. Int J Comput Appl 2013;81(16):40–7.

[10] Terzi DS, Terzi R, Sagiroglu S. Big data analytics for network anomaly detection from netflow data. In: Computer science and engineering (UBMK), 2017 international conference on. IEEE; 2017. p. 592–7. October.

[11] Dhanabal L, Shantharajah SP. A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. Int J Adv Res Comput Commun Eng 2015;4(6):446–52.

[12] Aggarwal P, Sharma SK. Analysis of KDD dataset attributes-class wise for intrusion detection. Procedia Comput Sci 2015;57:842–51.

[13] Deb K, Pratap A, Agarwal S, Meyarivan TAMT. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 2002;6(2):182–97.

[14] Ramachandran C. An advanced data processing based fusion IDS structures. Int J Appl Eng Res 2017;12(21):10929–37.

[15] S. Mirjalili, Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm, Knowl.-Based Syst. 89 (2015) 228–249.

[16] Abd Elaziz, M., Ewees, A.A., Ibrahim, R.A. and Lu, S., 2020. Opposition-based moth-flame optimization improved by differential evolution for feature selection. *Mathematics and Computers in Simulation*, *168*, pp.48-75.

[17] G.-B. Huang, Q. Zhu, C. Siew, G. H. Ã, Q. Zhu, C. Siew, G.-B. Huang, Q. Zhu, and C. Siew, “Extreme learning machine: Theory and applications,” Neurocomputing, vol. 70, no. 1–3, pp. 489–501, 2006.

[18] Y. Jian, D. Huang, J. Yan, K. Lu, Y. Huang, T. Wen, T. Zeng, S. Zhong, and Q. Xie, “A Novel Extreme Learning Machine Classification Model for eNose Application Based on the Multiple Kernel Approach,” 2017.

[19] Diker, A., Avci, D., Avci, E. and Gedikpinar, M., 2019. A new technique for ECG signal classification genetic algorithm Wavelet Kernel extreme learning machine. *Optik*, *180*, pp.46-55.

[20] Sarker, I.H., Abushark, Y.B., Alsolami, F. and Khan, A.I., 2020. Intrudtree: a machine learning based cyber security intrusion detection model. *Symmetry*, *12*(5), p.754.

[21] Maheswari, M., & Karthika, R. A. (2021). A Novel QoS Based Secure Unequal Clustering Protocol with Intrusion Detection System in Wireless Sensor Networks. Wireless Personal Communications, 118(2), 1535-1557.

[22] Lakshmanaprabu, S. K., Shankar, K., Ilayaraja, M., Nasir, A. W., Vijayakumar, V., & Chilamkurti, N. (2019). Random forest for big data classification in the internet of things using optimal features. International journal of machine learning and cybernetics, 10(10), 2609-2618.

[23] Metawa, N., Pustokhina, I. V., Pustokhin, D. A., Shankar, K., & Elhoseny, M. (2021). Computational Intelligence-Based Financial Crisis Prediction Model Using Feature Subset Selection with Optimal Deep Belief Network. Big Data, 9(2), 100-115.

[24] Lydia, E. L., Moses, G. J., Varadarajan, V., Nonyelu, F., Maseleno, A., Perumal, E., & Shankar, K. (2020). CLUSTERING AND INDEXING OF MULTIPLE DOCUMENTS USING FEATURE EXTRACTION THROUGH APACHE HADOOP ON BIG DATA. Malaysian Journal of Computer Science, 108-123.

[25] A. Muthumari, J. Banumathi, S. Rajasekaran, P. Vijayakarthik, K. Shankar et al., "High security for de-duplicated big data using optimal simon cipher," Computers, Materials & Continua, vol. 67, no.2, pp. 1863–1879, 2021.

[26] Lakshmanaprabu, S. K., Shankar, K., Rani, S. S., Abdulhay, E., Arunkumar, N., Ramirez, G., & Uthayakumar, J. (2019). An effect of big data technology with ant colony optimization based routing in vehicular ad hoc networks: Towards smart cities. Journal of cleaner production, 217, 584-593.