AN ENSEMBLE OF FEATURE SUBSET SELECTION WITH DEEP BELIEF NETWORK BASED SECURE INTRUSION DETECTION IN BIG DATA ENVIRONMENT

Gurrampally Kumar
Research Scholar, Department of Computer Science and Engineering,
Annamalai University Tamilnadu, India.
grk.040@gmail.com

Dr. S. Mohan
Department of Computer Science and Engineering, Annamalai University,
Annamalai Nagar, Chidambaram, Tamilnadu, India.
mohancseau@gmail.com

Dr A. Nagesh
Professor, Dept. of CSE, Mahatma Gandhi Institute of Technology, Hyderabad.
anagesh_cse@mgit.ac.in

Abstract - In recent times, massive quantity of data and its exponential rise have modified the significance of data security and analysis systems for Big Data. An intrusion detection system (IDS) is a process which helps to monitor and analyze the data for detecting the intrusions in the network. The huge quantity, variation, and high speed of data produced in the system pose a difficulty to the conventional techniques for the detection of attacks. Big Data methods are utilized in IDS for dealing with Big Data for precise and effective data analytic processes. This paper introduces a novel feature subset selection (FSS) with deep belief network (DBN) based intrusion detection in big data environment, called FSS-DBN model. The presented model involves data preprocessing stage to improve the data quality to a certain extent. In addition, the FSS can be considered as an optimization problem and is effectively solved by the use of teaching and learning based optimization (TLBO) algorithm. Moreover, DBN model is applied for identifying the class labels (i.e. intrusions) in the network. For validating the proficient results analysis of the FSS-DBN model, an extensive set of simulations were performed and the superior performance is also highlighted with the maximum sensitivity of 0.9898, specificity of 0.9865, accuracy of 0.9854, F-score of 0.9872, and kappa of 0.9867.

***Keywords*:** Big data, security, intrusion detection, feature selection, deep learning, TLBO algorithm.

1. Introduction

Big Data refers to the massive quantity of data which makes it hard to save, manage and investigate it by the use of classical databases and software models. It encompasses huge volume and velocity, and also variation of data which necessitate novel models for handling it. An intrusion detection system (IDS) is tool commonly used for monitoring purposes which examines the data for the detection of attacks toward the network. Conventional IDS models resulted in a complicated and ineffective system in managing big data, due to the fact that it examines the properties process is difficult and requires maximum duration. The longer duration of data examination poses for particular duration prior to receiving alerts [1]. So, Big Data tools and techniques can be used for analyzing and storing data in the IDS reducing the computational and training time. The IDS involves 3 distinct ways to detect attacks namely Signature, Anomaly, and Hybrid based detection models.

The former model is developed for the detection of known attacks utilizing the signature of the attack. It is a proficient model of identifying the known attacks which already existed in the IDS databases. So, it is found to be a novel way of detecting intrusions in case of known attacks [2]. But they have failed to identify new attacks which do not exist in the signature and the database needs to be adequately updated to improve the detection rate [3]. For resolving this issue, Anomaly based detection methods perform a comparison of the present client actions over the fixed profile for detecting abnormalities which may be considered an intrusion. It is useful to detect unknown or zero day attacks with no update to the system. But this technique normally has maximum false

positive rate (FPR) [4, 5]. The final hybrid methods involve integration of two or many IDS for addressing the drawbacks of the single technique utilized and attain the benefits of multiple techniques involved. Several researchers have employed machine learning (ML) and deep learning (DL) models to detect intrusions with high detection rates. But for dealing with big data, the ML algorithms require maximum duration to learn and classify data. The utilization of big data and ML models can resolve different issues namely speed and processing time to design precise IDS. The major aim of this paper is to employ Big Data approaches for dealing with the big data in IDS for reducing the processing time and attains proficient classification.

[6] defined the whole task to handle and manage big data. They have carried out a comprehensive and useful examination over the huge data, embed Hadoop MapReduce model for incorporating the rapid computation process in big data by the use of different ML models and also utilize clustering techniques for reducing load. [7] presented a technique reflecting the benefits of utilizing fusion based pattern recognition technique for detecting intrusions in the network by the use of many classification models. [8] highlighted the noteworthy and effectual usage of Apache Spark to process huge amount of data with the idea of parallelism which assists rapid computation of big data.

[9] developed an enhanced model on Spark system, which utilizes feature subset selection (FSS) with random forest (RF) classification model on the UCI dataset. Two major processes involved in the FSS comprising the reduction of noisy features and eliminating repetitive features. [10] verified the advantages of exemplifying NSGA-II for an improved FSS model that is an extended model of the available NSGA model by considering the challenges in NSGA includes maximum computation complexity, requiring sharing parameter specification, and absence of elitism. [11] developed an IDS model for the generation of rules by the use of NSGA-II which makes use of multi-objective technique. In addition, DARPA database is applied for evaluating the simulation outcome which considerably results in speed up the rule creation process and elevates the secrecy in the rule based IDS model.

[12] make use of KDD cup 99 datasets and assumed the significant features out of the available 41 features results in the reduction in efficiency of the IDS based on accuracy and detection rate. Therefore, NSL KDD cup 99 dataset is used as an extended KDD cup 99 dataset to effectively detect novel attacks on computer systems since no repetitive record or missing value existed. [13] presented a model which carries out the fusion of many IDS comprising signature and anomaly based techniques. The experimental values indicated that the efficiency can be enhanced by reducing the FPR through the fusion of multiple models. [14] designed a taxonomy of IDS with data mining, FSS, and preprocessing on KDDCup 99 dataset. In addition, the preprocessing and FSS processes take place at the learning and testing stages respectively. The experimental values showcased that the fusion model offers improved efficiency and increased false positive rates.

This paper introduces a novel feature subset selection (FSS) with deep belief network (DBN) based intrusion detection in big data environment, called FSS-DBN model. The presented model involves data preprocessing stage to improve the data quality to a certain extent. In addition, the FSS can be considered as an optimization problem and is effectively solved by the use of teaching and learning based optimization (TLBO) algorithm. Moreover, DBN model is applied for identifying the class labels (i.e. intrusions) in the network. For validating the proficient results analysis of the FSS-DBN model, an extensive set of simulations were performed and the superior performance is also highlighted in terms of distinct measures.

2. The Proposed FSS-DBN based intrusion detection model

Fig. 1 demonstrates the processes that exist in the FSS-DBN based intrusion detection model. The figure defined that the networking data are initially preprocessed to transform it to a useful format. Then, the optimal feature subsets are chosen through the application of TLBO algorithm. Next, the DBN based classifier gets executed to determine the existence of intrusions in the network. The proposed system entities are discussed as follows.

• Hadoop MapReduce—Hadoop is a commonly available “Big Data Handler” and it is mainly developed for managing big data. Hadoop 2.6.0 version is used for simulation purposes. In addition, Hadoop defines the usage of distinct technologies namely HDFS, MapReduce, HBase, Hive, Pig, and so on. In Hadoop, YARN is accountable to manage every integrated resource and job scheduling task. In particular cases, the data is required instantly and is saved in HDFS. The hadoop is saved data in peta-bytes and zetabytes with no restriction on storing, and if utilized in collaboration by MapReduce afterward gives faster calculation and modeling of information [15]. MapReduce is a batch-oriented programming system in Hadoop which carries out organization information as well as scheduling of jobs. The MapReduce initial separates data as independent chunks which are treated completely with map roles in corresponding and behind sorted with structure. Afterward, the result is provided as input to diminished the role. The Hadoop and MapReduce cooperatively succeed data. Fig. 2 illustrates the components involved in Apache Spark.

• Apache Spark—Spark is a BDS made on Hadoop and runs on Hadoop Yarn. It can be utilizing Apache Spark version 2.1.0. The spark is standalone or able to effort from collaboration to structure of MapReduce and HDFS, beyond that is utilized later. In spark's modules contain SQL, Streaming, and MLlib (ML library). The classifiers in Apache Spark are until progressing. Hadoop clusters are able to run collaborating queries, streaming data, and many further on Apache Spark.

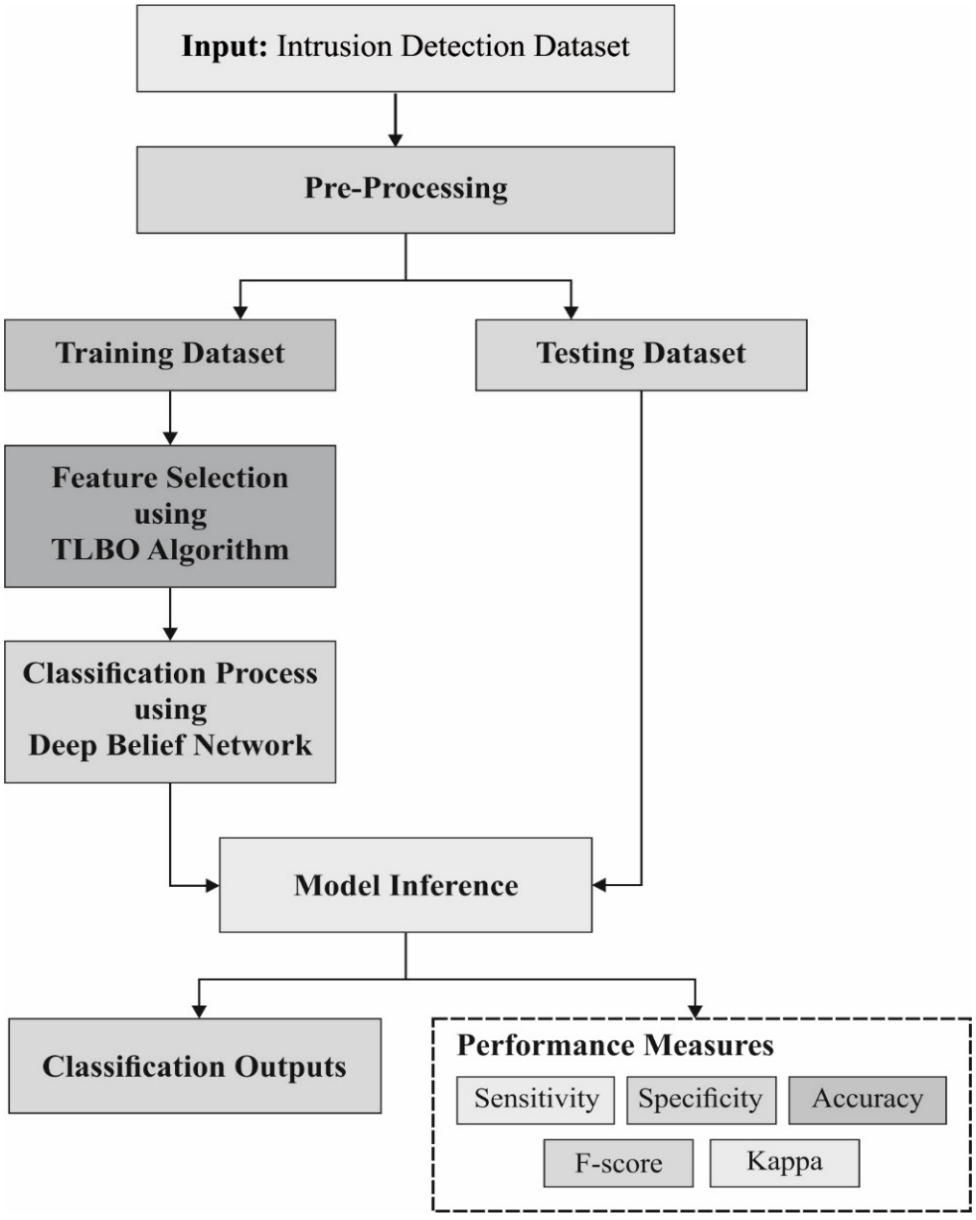

Fig. 1. Working process of FSS-DBN model.

• Spark RDD—The spark keeps a programming method which is much related to MapReduce but extends it through "Resilient Distributed Datasets" (RDD) which is a data-sharing abstraction. Utilizing this extension, Spark is capable to show a wider range of modeling capabilities that before required distinct engines, containing streaming, ML, SQL, and graph modeling.

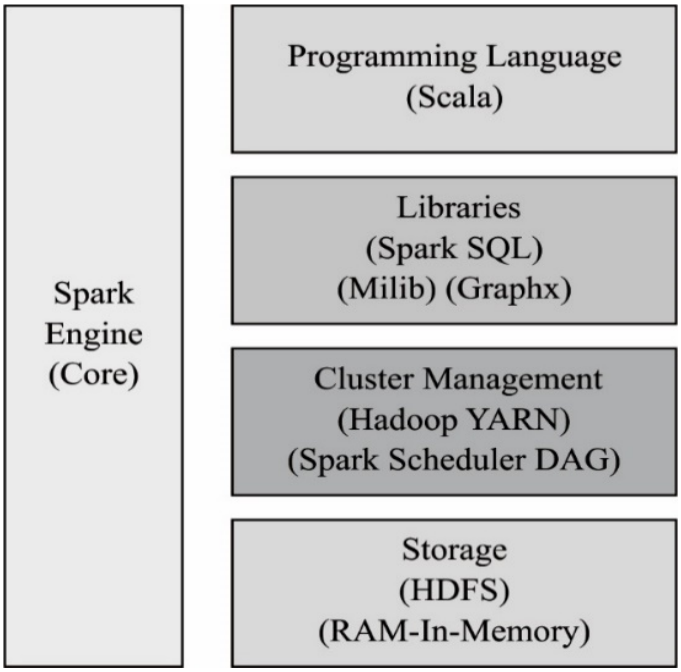

Fig. 2. Apache Spark Components.

2.1. *TLBO based FS Model*

The TLBO technique is based on the idea of knowledge sharing amongst the teachers and students in the learning procedure. It depends upon the influence of the teacher and class efficiency. A set of two phases have existed in the TLBO algorithm namely 'Teacher' and 'Training' phases. The characteristics of the effectual teacher tried to improve the understandability to a certain level. In practical cases, it is difficult and the teacher can attain the average class to a specific label based on distinct ways. For instance, M_i denotes the mean value of the class and T_i designates teacher at iteration i. In addition, the teacher T_i has tried to move in the way of mean M_i closer to the individuals, and the updated mean T_i can be represented as called as M_{new}. The solution gets updated based on the variation that exists among the current and updated mean M_{new} is given in Eq. (1).

$$\text{Difference\_Mean}_i = r_i(M_{new} - T_F M_i) \quad (1)$$

where T_F indicates a teaching factor which computes the mean value to be modified and r_i is an arbitrary number that existed in the range of [0, 1]. The T_F value can be defined as 1 or 2, which is chosen randomly as given below.

$$TF = round\,[1 + rand\,(0,1)\{2 - 1\}] \quad (2)$$

This difference gets modified the existing solution as listed below.

$$Z_{new,\ i} = Z_{old,\ i} + \text{Difference\_Mean}_i \quad (3)$$

Learners get improved by the knowledge by utilizing the two methods such as follows.

The initial one receives the input from teacher and second one interacted with one another. A learner enhanced the knowledge by random interaction with other learners. In general, the knowledge of the learner gets interacted with the highly informed learner. In those situations, the learner update is defined here. through interacting randomly with other learners. Generally, the knowledge of a learner interacts with the more well-informed learner [16]. In such cases, the learner modification is applied as follows.

$For\ i = 1 :\ P_n$

Arbitrary selection of two learners A_i and A_j, where i≠j

$If\ f(A_i) < f(A_j)$

$A_{n,i} = A_{o,i} + r_i(A_i - A_j)$

Else

$A_{n,i} = A_{o,i} + r_i(A_j - A_i)$

End If

End For

Enable Z_n if an improved function is obtained.

The TLBO-FSS model enables to modify of the classical TLBO algorithm for searching the solution space to elect an optimum subset of features. Fig. 3 demonstrates the flowchart of TLBO technique. The processes get repeated until the termination criteria get reached, i.e., predefined iteration count. The optimal learners denote the optimal solutions in a specific round and the procedure gets continued with increasing number of iterations. The processes involved in the TLBO-FSS model are defined below. Here, the learners undergo training using the teacher. At every round k, assume that there is 's' feature count $\{f = 1,2 \dots s\}$, 't' number of samples (i.e., population $\{i = 1,2 \dots t\}$). The steps involved in the TLBO-FSS model are given as follows.

Step 1: Parameter initialization: Population, Feature count as $Z_{f,i,k}$ and stopping criteria.

Step 2: Determine the average of every individual feature for the learners as $M_{f,k}$.

Step 3: Determine the fitness value of every individual as given below.

$$Fitness(Z_{f,i,k}) = Accuracy(Z_{f,i,k}) \quad (4)$$

Step 4: Updating the learner using the teacher (*Teacher Phase*)

a) Elect an optimal learner in the population as Teacher.

b) Determine the variation mean of every attribute based on the fittest individual as given below.

$$Diff_{M}ean_{f,i,k} = r_k(Z_{f,ibest,k} - T_F M_{f,k}) \quad (5)$$

where, $Z_{f,ibest,k}$ is the optimal set of individuals in subject f. TF is the teaching factor by the values of 1 or 2 and r_k is an arbitrary number that lies in the interval of $[0, 1]$.

c) The optimal set of individuals acted as teachers and train the residual individuals. Every individual learner in the population gets updated by use of Eq. (6).

$$Z'_{f,i,k} = 0 \; if \; Z_{f,i,k} + Diff_{Mean}f,k < 0.5$$
$$Z'_{f,i,k} = 1 \; if \; X_{f,i,k} + Diff_{Mean}f,k \geq 0.5 \quad (6)$$

where, $Z'_{f,i,k}$ is the trained value of $Z_{f,i,k}$.

d) If $Z'_{f,i,k}$ exceeds $Z_{f,i,k}$,

Remain the previous value

Else,

Replace with new value.

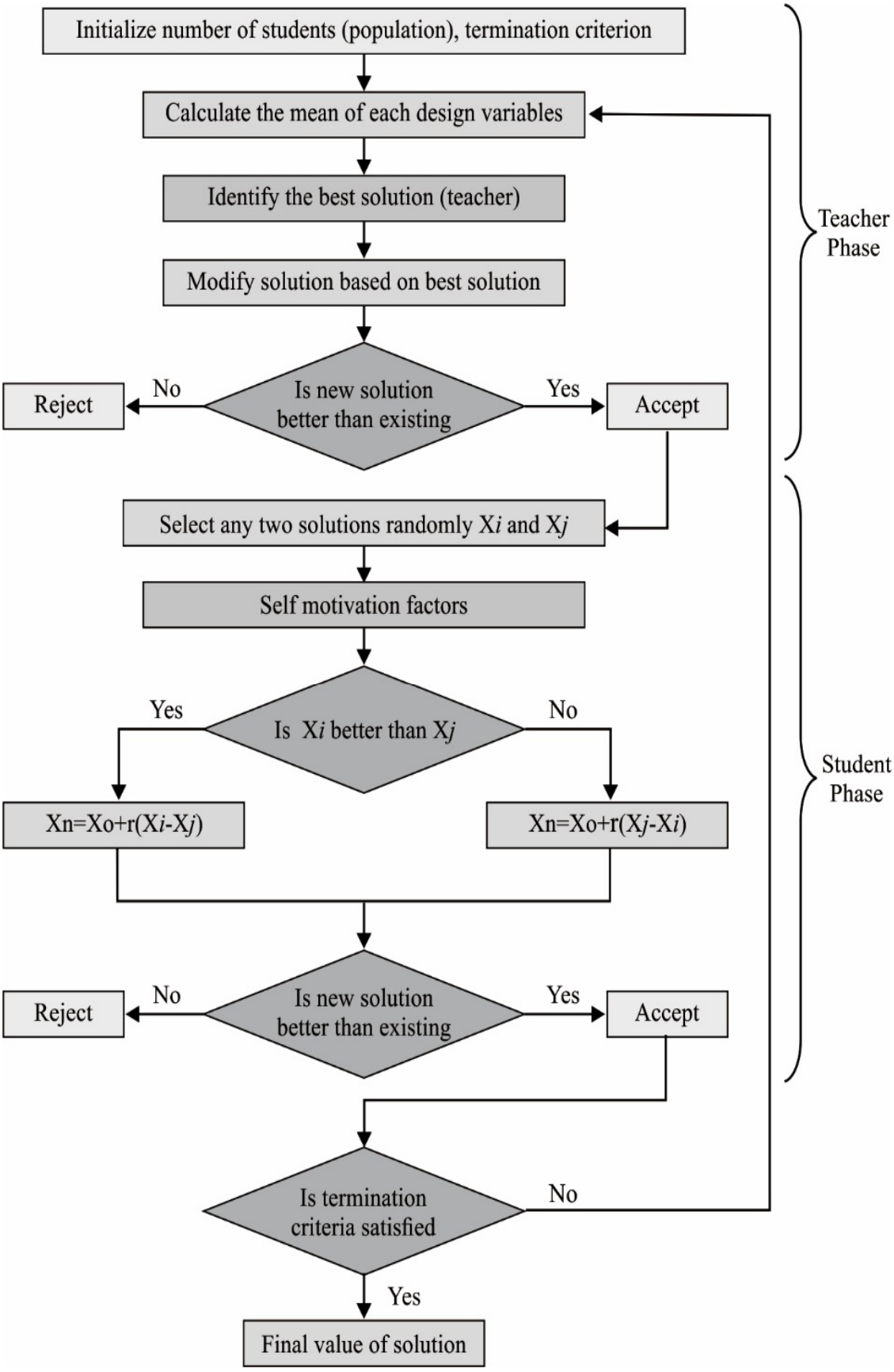

Fig. 3. Flowchart of TLBO.

Step 5: Upgrade the learners by the use of another learner through Eqs. (7) and (8) (*Learner Phase*)

a) Elect 2 set of sample U and V with the criteria $Z'_{total-U,k} \neq Z'_{total-V,k}$ randomly.

where, $Z'_{total-U,k}$, $Z'_{total-V,k}$ are restructured features of $Z_{total\text{-}U,k}$, $Z_{total-V,k}$ of U and V correspondingly.

b) If $Z'_{total-U,k}$ exceeds $Z'_{total-V,k}$

$$Z''_{f,U,k} = 0 \; if \; Z'_{f,U,k} + r_k\left(Z'_{f,U,k} - Z'_{f,V,k}\right) < 0.5$$
$$Z''_{f,U,k} = 1 \; if \; Z'_{f,U,k} + r_k\left(Z'_{f,U,k} - Z'_{f,V,k}\right) \geq 0.5 \quad (7)$$

Else,

$$Z''_{f,U,k} = 0 \; if \; Z'_{f,U,k} + r_k\left(Z'_{f,V,k} - Z'_{f,U,k}\right) < 0.5$$
$$Z''_{f,U,k} = 1 \; if \; Z'_{f,U,k} + r_k\left(Z'_{f,V,k} - Z'_{f,U,k}\right) \geq 0.5 \quad (8)$$

c) If $Z''_{f,U,k}$ exceeds $Z'_{f,U,k}$

Afterward, endure with the previous value

Else,

Replace with the prior value.

Step 6: When the stopping criteria get satisfied,

Display the outcome

Else, jump to Step 2

2.2. *DBN based Classification*

Next to the FSS process, DBN model is applied. It depends upon the Restricted Boltzmann Machine (RBM) comprises visible and hidden layers; the neurons amongst 2 layers are fully connected (FC), and the neurons in the identical layer are disconnected. Here, $v(v_1, v_2, \ldots, v_n)$ denotes the visible layer, v_i is the visible unit; $h(h_1, h_2, \ldots, h_m)$ defines the hidden layer, h_j is the hidden unit, and W is the connection weight matrix amongst the 2 layers. The data is fed as input from the visible layer, $(v_1, v_2, \ldots, v_n)$ denotes the feature group of data, and hidden layer data is produced with an arbitrarily initializing the weight value *w* and the state of every individual neuron.

Provided a collection of states (v, h), the energy function of the RBM method is represented as follows:

$$E(v,h) = -\sum_{i=1}^{n} a_i \, v_i - \sum_{j=1}^{m} b_j \, h_j - \sum_{i=1}^{n}\sum_{j=1}^{m} v_i \, w_{ij} h_j, \quad (9)$$
$$= -a^T v - b^T h - h^T w v,$$

where $a = (a_1, a_2, \cdots, a_n)$ and $v = (v_1, v_2, \ldots, v_n)$ indicates the offset and state vector of the visible unit, $b = (b_1, b_2, \cdots, b_m)$ and $h = (h_1, h_2, \ldots, h_m)$ denotes the is the bias and state vector of the hidden layer, $w = \left(w_{i,j}\right)$ is the connection weight matrix, and $w_{i,j}$ represents the weight of *i*th visible and *j*th hidden elements. By the states of (v, h), based on Eq. (9), the joint probability distribution is represented by:

$$P(v, h;\, \theta) = \frac{1}{z} e^{-E(v,h)}, \quad (10)$$
$$z = \sum_{v}\sum_{h} e^{-E(v,h)},$$

where $\theta = \{a, b, w\}$ is the RMB network variables and Z is known as the normalization factor or partition function. Practically, the probability distribution $p(v)$ of training dataset v is commonly utilized, i.e., the edge probability distribution of $P = (v, h, \theta)$ is defined as follows.

$$P(v) = \frac{1}{Z}\sum_{h} e^{-E(v,h)}. \quad (11)$$

Likewise, the edge probability distribution $P = (h)$ of the hidden layer state is attained as follows.

$$P(h) = \frac{1}{Z}\sum_{v} e^{-E(v,h)}. \quad (12)$$

RBM training dataset is attained by resolving the optimum variables, therefore, the method is effectively suitable for the distribution of training dataset even when the instances reach the highest probability in the distribution. The designing of log- probability function is given by:

$$\ln P(v) = \ln\left(\sum_{h} e^{-E(v,h)}\right) - \ln\left(\sum_{v}\sum_{h} e^{-E(v,h)}\right). \quad (13)$$

The model variables are correspondingly resolved using the highest probability function technique:

$$\frac{\partial \ln P(v)}{\partial a} = E_{\mathrm{Pd}}[v] - E_{\mathrm{Pm}}[v],$$
$$\frac{\partial \ln P(v)}{\partial b} = E_{\mathrm{Pd}}[h] - E_{\mathrm{Pm}}[h], \quad (14)$$
$$\frac{\partial \ln P(v)}{\partial w} = E_{\mathrm{Pd}}[vh^T] - E_{\mathrm{Pm}}[vh^T],$$

where E_{Pd} and E_{Pm}signifies the expectation of an input conditional and joint probability distribution of training dataset [17]. In addition, the processing is carried out using the Gibbs sampling technique whereas the processing is high at every iteration. If the state of neurons in the input layer is provided, the activation probability of the hidden unit is defined by

$$P(h_j = 1|v) = \sigma\left(\sum_{i} v_i\, w_{ij} + b_j\right). \quad (15)$$

Next to determining the hidden component of the state matrix, the reconfigurable visible component state probability is determined using Eq. (16):

$$P(v_j = 1|h) = \sigma\left(\sum_{j} h_j\, w_{ij} + a_i\right), \quad (16)$$

where σ is a sigmoid function $\sigma(x) = 11/(1 + \exp(-x))$.

The highest possibility function value is continually approached using gradient ascent and the variable in the RBM model is updated using Eq. (17):

$$\theta^{i+1} = \theta^i + \eta\, \frac{\partial\, lnP(v)}{\partial \theta}, \quad (17)$$

where η is the parameter learning rate, and i is the present round. The variable θ gets repeatedly updated based on Eq. (17). The DBN comprises many RBM components which are linked to the end layer of RBM visible and input layers. The RBM is a probabilistic neural network (PNN) which computes the probability generation of DBN defining the joint possibility distribution amongst the feature and labels:

$$P(v, h_1, h_2, \cdots, h_l) = P(v|h_1)P(h_1|h_2), \ldots, P(h_{l-2}|h_{l-1})P(h_{l-1}|h_l), \quad (18)$$

where $P(h_k|h_{k+1})$ is the conditional possibility distribution of h_k to provided h_{k+1} state; $P\ (h_{l-1}, h_l)$ denotes the joint likelihood distribution of h_{l-1} and h_l. $P(v, h)$ refers the joint likelihood distribution of an individual RBM. The utilization of the DBN establishes a DL based IDS method for achieving accurateness and stableness of the predictive techniques.

3. Performance Validation

The experimental results of the FSS-DBN model are validated using benchmark KDDCup 1999 dataset, which includes a total of 125973 instances with a set of 41 features with two classes. In addition, a set of 67343 instances come under 'Normal' class and a total of 58630 instances come under 'Abnormal' class as shown in Table 1.

Table 1. Dataset Description.

Source	# of instances	# of attributes	# of class	Normal/Anomaly
KDD Cup 1999	125973	41	2	67343/58630

Table 2 and Fig. 4 investigates the FSS outcomes of the TLBO-FSS model with other existing methods interms of best cost. The obtained values indicated that the BGOA-V algorithm has failed to achieve effective FSS and achieved the highest best cost of 0.006763. Also, the BGOA technique has accomplished slightly improved FSS outcomes by offering the best cost of 0.006530. Followed by, the BGOA-S model has accomplished moderate FSS of 0.004176. Though the GA-FS model has chosen features with the near optimal best cost of 0.001150, the presented TLBO-FSS model has resulted in an effective FSS performance with the best cost of 0.001108. In addition, the TLBO-FSS model has chosen a total of 21 features out of 41 features.

Table 2. Comparative studies of existing feature selection with presented TLBO-FSS method for Applied Dataset.

Methods	Best Cost	Selected Features
TLBO-FSS	0.001108	2,4,5,6,8,9,11,13,14,16,18,19,22,24,25,29,30,32,37,39,40
GA-FS	0.001150	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
BGOA-S	0.004176	3,5,8,13,18,20,21,22,30,32,33,34,36,38,40,2,7,19,23,35,27,29
BGOA-V	0.006763	21,27,32,25,34,1,35,3,24,40,28,26,10,5,33,14,16,12,36,23,30,38,22,15,37,9
BGOA	0.006530	23,34,35,36,23,29,21,37,2,1,6,7,9,11,15,19,20,22,27,39,40,3,5

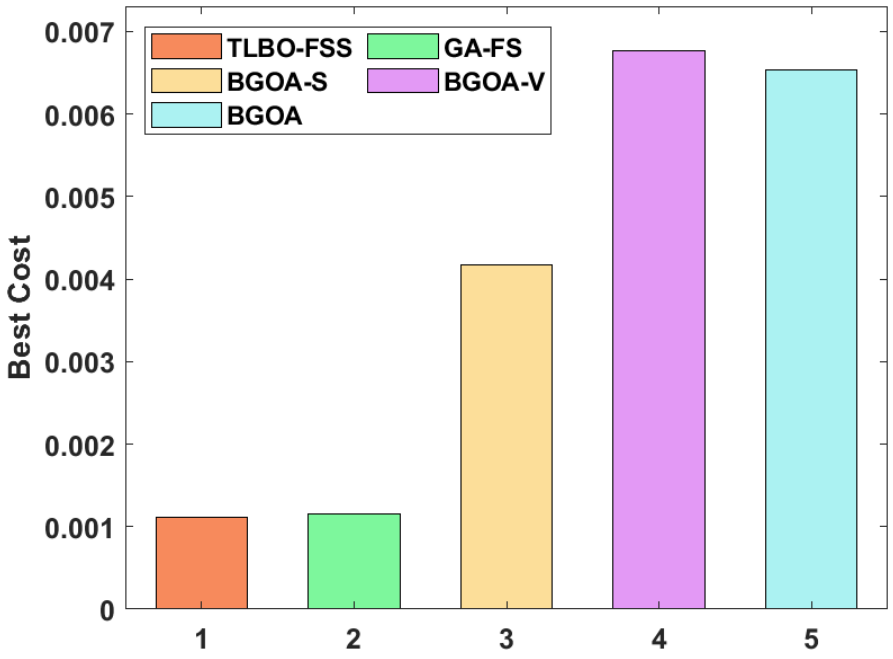

Fig. 4. Feature selection of TLBO-FSS model interms of best cost.

Table 3 provides a detailed comparative study of the FSS-DBN model with other existing iDS models in big data environment.

Fig. 5 examines the accuracy analysis of the FSS-DBN model with compared methods. The figure demonstrated that the Fusion model has obtained the lowest accuracy of 0.9203 whereas a slightly increased accuracy of 0.9293 has been offered by the RBFNetwork model. Besides, the RF model has tried to showcase moderate accuracy of 0.9304 and the DT model has depicted somewhat reasonable accuracy of 0.9553. At the same time, the RT model has exhibited manageable accuracy of 0.9555 and the LR model has accomplished a certainly improved accuracy of 0.974. Moreover, a competitive outcome is exhibited by the DBN model with an accuracy of 0.974. However, the presented FSS-DBN model has resulted in a maximum accuracy of 0.9854.

Table 3. Performance Evaluation of Existing Models with Proposed FSS-DBN Model on Intrusion Detection System Dataset.

Methods	Sensitivity	Specificity	Accuracy	F-score	Kappa
FSS-DBN	0.9898	0.9865	0.9854	0.9872	0.9867
DBN	0.9892	0.9743	0.9740	0.9732	0.9729
Fusion Model	0.9146	0.9221	0.9203	0.9190	0.9130
RBFNetwork	0.9340	0.9238	0.9293	0.9338	0.8579
LR	0.9726	0.9692	0.9710	0.9729	0.9419
Random Forest	0.9239	0.9383	0.9304	0.9358	0.8599
Random Tree	0.9568	0.9539	0.9555	0.9584	0.9106
Decision Tree	0.9568	0.9537	0.9553	0.9583	0.9103

Fig. 6 investigative the sensitivity and specificity analysis of the FSS-DBN method with related techniques. The figure exhibited that the Fusion approach has attained a minimum sensitivity of 0.9146 whereas a somewhat higher sensitivity of 0.9239 has been offered by the RF technique. Likewise, the RBFNetwork manner has tried to depicted moderate sensitivity of 0.9340 and the DT and RT methodologies have showcased somewhat increased and similar sensitivity of 0.9568. Simultaneously, the LR model has outperformed manageable sensitivity of 0.9726. In addition, a competitive outcome is portrayed by the DBN model with a sensitivity of 0.9892. But, the projected FSS-DBN model has resulted in a superior sensitivity of 0.9898.

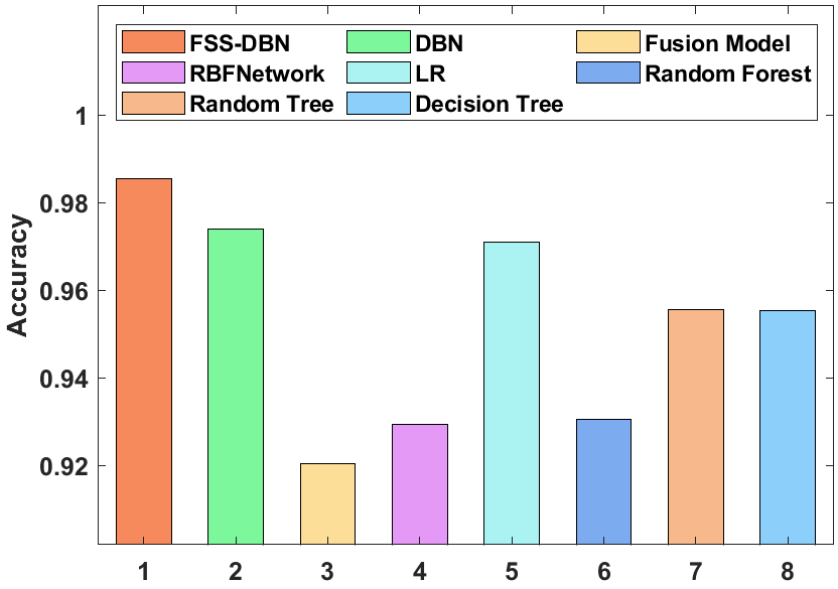

Fig. 5. Accuracy analysis of FSS-DBN model on IDS dataset.

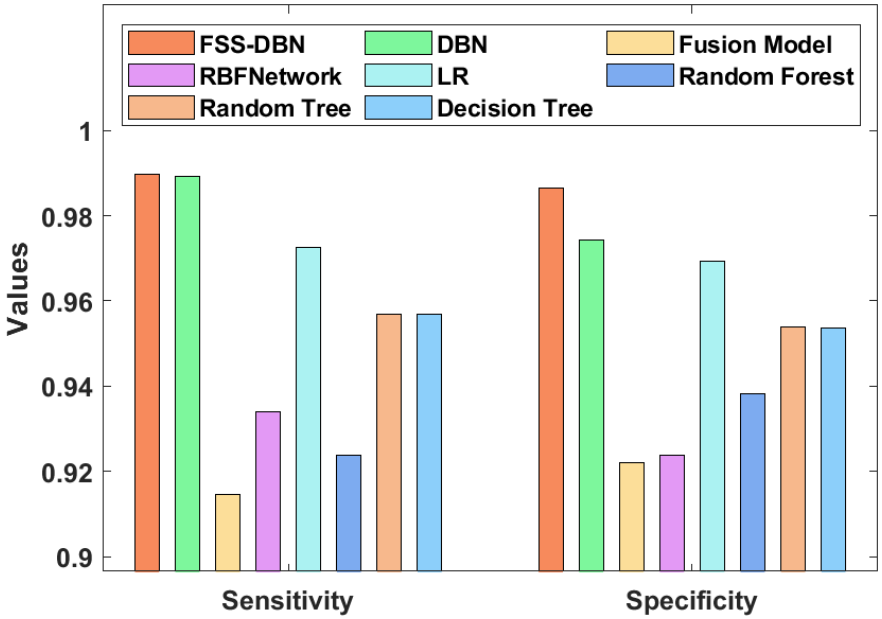

Fig. 6. Sensitivity and specificity analysis of FSS-DBN model on IDS dataset.

The figure portrayed that the Fusion approach has reached a lower specificity of 0.9221 whereas a slightly improved specificity of 0.9238 has been offered by the RBFNetwork model. In line with, the RF approach has tried to exhibit moderate specificity of 0.9383 and the DT algorithm has demonstrated slightly reasonable specificity of 0.9537. Along with that, the RT technique has showcased manageable specificity of 0.9539 and the LR approach has accomplished a certainly higher specificity of 0.9692. Followed by, a competitive outcome is outperformed by the DBN model with the specificity of 0.9743. Finally, the presented FSS-DBN technique has resulted in a higher specificity of 0.9865.

Fig. 7 showcases the F-score and Kappa analysis of the FSS-DBN model with compared algorithms. The figure exhibited that the Fusion technique has attained the lowest F-score of 0.9190 whereas a somewhat higher F-score of 0.9338 has been offered by the RBFNetwork model. Similarly, the RF technique has tried to outperform moderate F-score of 0.9358 and the DT technique has demonstrated somewhat reasonable F-score of 0.9583. Concurrently, the RT model has depicted manageable F-score of 0.9584 and the LR method has accomplished a certainly improved F-score of 0.9729. Additionally, a competitive result is exhibited by the DBN model with an F-score of 0.9732. Eventually, the presented FSS-DBN model has resulted in a maximum F-score of 0.9872.

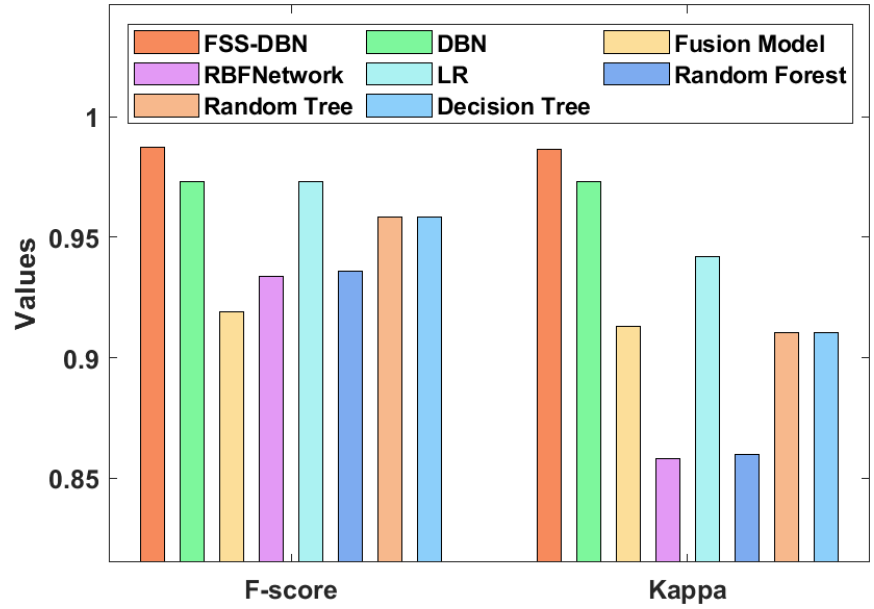

Fig. 7. F-score and Kappa analysis of FSS-DBN model on IDS dataset.

The figure showcased that the RBFNetwork model has achieved a minimal kappa of 0.8579 whereas a slightly higher kappa of 0.8599 has been offered by the RF approach. Also, the DT method has tried to illustrate moderate kappa of 0.9103 and the RT manner has depicted somewhat reasonable kappa of 0.9106. On continuing with, the Fusion technique has portrayed manageable kappa of 0.9130 and the LR technique has accomplished a certainly increased kappa of 0.9419. In addition, a competitive result is outperformed by the DBN technique with the kappa of 0.9729. At last, the projected FSS-DBN algorithm has resulted in a superior kappa of 0.9867.

4. Conclusion

This paper has devised a novel FSS-DBN model for intrusion detection in big data environment. The presented model involves data preprocessing stage to improve the data quality to a certain extent. In addition, the FSS can be considered as an optimization problem, and the optimal feature subsets are chosen through the application of TLBO algorithm. Next, the DBN based classifier gets executed to determine the existence of intrusions in the network. For validating the proficient results analysis of the FSS-DBN model, an extensive set of simulations were performed and the superior performance is also highlighted in terms of distinct measures. The experimental values ensured that the FSS-DBN model has resulted in a maximum sensitivity of 0.9898, specificity of 0.9865, accuracy of 0.9854, F-score of 0.9872, and kappa of 0.9867. As a part of future scope, the presented FSS-DBN model can be extended to the application of clustering and outlier removal techniques.

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] Sahasrabuddhe A, et al. Survey on intrusion detection system using data mining techniques. Int Res J Eng Technol. 2017;4(5):1780–4.
[3] 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.
[4] Scarfone K, Mell P. Guide to intrusion detection and prevention systems (idps). NIST Spec Publ. 2007;2007(800):94.
[5] Debar H. An introduction to intrusion-detection systems. In: Proceedings of Connect, 2000. 2000.
[6] Dietrich D, Heller B, Yang B. Data science & big data analytics: discovering. Analyzing, visualizing and presenting data; 2015.
[7] Giacinto G, Roli F, Didaci L. Fusion of multiple classifiers for intrusion detection in computer networks. Pattern Recog Lett 2003;24(12):1795–803.
[8] Zaharia M, Xin RS, Wendell P, Das T, Armbrust M, Dave A, et al. Apache spark: a unified engine for big data processing. Commun ACM 2016;59(11):56–65.
[9] Sun K, Miao W, Zhang X, Rao R. An improvement to feature selection of random forests on spark. In: Computational science and engineering (CSE), 2014 IEEE 17th international conference on. IEEE; 2014. p. 774–9.
[10] 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.
[11] Tamimi A, Naidu DS, Kavianpour S. An Intrusion detection system based on NSGA-II Algorithm. In: Cyber security, cyber warfare, and digital forensic (CyberSec), 2015 fourth international conference on. IEEE; 2015. p. 58–61. October.
[12] Sung AH, Mukkamala S. Identifying important features for intrusion detection using support vector machines and neural networks. In: Applications and the internet, 2003. Proceedings. 2003 symposium on. IEEE; 2003. p. 209–16.
[13] Mulay SA, Devale PR, Garje GV. Intrusion detection system using support vector machine and decision tree. Int J Comput Appl 2010;3(3):40–3.
[14] Ramachandran C. An advanced data processing based fusion IDS structures. Int J Appl Eng Res 2017;12(21):10929–37.
[15] Donkal, G. and Verma, G.K., 2018. A multimodal fusion based framework to reinforce IDS for securing Big Data environment using Spark. Journal of information security and applications, 43, pp.1-11.
[16] Allam, M. and Nandhini, M., 2018. Optimal feature selection using binary teaching learning based optimization algorithm. Journal of King Saud University-Computer and Information Sciences.
[17] Lu, P., Guo, S., Zhang, H., Li, Q., Wang, Y., Wang, Y. and Qi, L., 2018. Research on improved depth belief network-based prediction of cardiovascular diseases. Journal of healthcare engineering, 2018.