Intelligent Differential Evolution based Feature Selection with Deep Neural Network for Intrusion Detection in Wireless Sensor Networks

Ibrahim M. EL-Hasnony

Faculty of Computers and Information, Mansoura University, Egypt

Emails: ibrahimhesin2005@mans.edu.eg

Abstract

A wireless sensor network (WSN) is mainly utilized for data gathering and surveillance applications. As WSN is majorly deployed in harsh and hostile environments, security remains a critical issue that needs to be resolved. An intrusion detection system (IDS) is one of the proficient ways used to determine the presence of abnormal behaviors (i.e. intrusions) in the network. Earlier studies have focused on the design of machine learning (ML) and deep learning (ML) models to design IDS. With this motivation, this paper presents an intelligent differential evolution-based feature selection with a deep neural network (IDEFS-DNN) for intrusion detection in WSN. The proposed IDEFS-DNN model aims to select the optimum set of features and classify the intrusions in the network. In addition, the IDEFS-DNN technique involves the design of the IDEFS technique to choose a subset of optimum features. Moreover, the chosen features are fed into the DNN technique for classification purposes. The usage of the IDEFS technique helps to reduce the complexity and increase the classifier outcome. To portray the improved performance of the IDEFS-DNN technique, wide-ranging experiments take place on benchmark datasets and the results are inspected under varying aspects. The simulation results ensured the enhanced intrusion detection performance of the IDEFS-DNN technique over the other IDS models.

Keywords: Security; WSN; Intrusion detection; Deep neural network; Feature selection; Metaheuristics.

1. Introduction

A wireless sensor network (WSN) is comprised of numerous sensor nodes positioned in regions where the goal is to gather information and transmit it for analysis. It is increasingly becoming an attractive area of research in resolving thus Real-time challenges, like military applications, environmental monitoring [1], traffic control, home automation, and geographical sensing. The WSN property shows that sensors are fully controlled by the resources, involving, bandwidth, memory energy, computing, and communication [2]. Consequently, the placement of this type of network using their resource limitations make their security question important, and susceptible to many security problems. WSN is extremely susceptible to attacks, because of their distributed and open nature and constrained resources of the sensors. Furthermore, in WSN packets transmitting should be frequently performed, sensors are placed arbitrarily in an environment hence attackers are injected easily into a WSN [3]. Denial of Service (DoS) attacks are dangerous and widespread attacks that threaten WSN privacy. These attacks have many forms and their primary aim is to suspend/interrupt the service given by WSN [4]. Since the procedure of preventing/avoiding security risks could not often effective, an Intrusion Detection System (IDS) is required for detecting known as well as unknown attacks and alert sensors [5]. IDS enables

DOI: https://doi.org/10.54216/JISIoT.000204

abnormal activities/detection of suspicious and triggers an alarm while an intrusion takes place. The execution of IDS for WS is very complex when compared to another scheme since sensors are generally developed to be cheap and, tiny and they don't have sufficient hardware resources. Furthermore, there are no particular datasets that comprise standard profiles and attacks in WSN which is utilized for detecting an attacker signature [6]. Considering the abovementioned problems, there are 2 criteria when developing IDS for WSN: The IDS should be of a higher amount of precision in intruder detection which comprises unknown attacks, and it should be lightweight for ensuring minimal overhead on the framework of WSN.

Moreover, an IDS is categorized according to the detection method. Several detection methods are presented in the area of IoT and WSN [7]. They utilize misuse detection approaches (denoted as a signature- or pattern-based), however, most of them employ anomaly detection. The misuse recognition technique distinguishes known attacks with higher success rates but has noteworthy disadvantages as it needs the previous experience of the attacks and their signature/pattern [8]. Alternatively, the anomaly detection technique distinguishes new attacks, therefore reducing the FNR, but, has additional false positive alarms. It needs offline training for determining the standard behaviors of the networks and the settings of specific limits/markers, which characterizes the usual behaviors. On the positioning, network action is linked to the present threshold and slight deviations from what is considered standard are categorized as anomalous.

Setting the threshold very higher might increase higher false negative alerts while setting the thresholds very lower might increase higher false positive alerts. Malicious behaviors are determined as the network behaviors generated by compromised nodes using the aim to disrupt/compromise WSN goals [9]. Frequently Attack targets network layer susceptibilities. Blackhole attacks and Selective. Nevertheless, the network layer they are directing also affects another local sensor network layer [10].

This paper presents an intelligent differential evolution-based feature selection with a deep neural network (IDEFS-DNN) for intrusion detection in WSN. The proposed IDEFS-DNN model aims to select the optimum set of features and classify the intrusions in the network. In addition, the IDEFS-DNN technique involves the design of the IDEFS technique to choose a subset of optimum features. Moreover, the chosen features are fed into the DNN model for classification purposes. The usage of the IDEFS technique helps to reduce the complexity and increase the classifier outcome. To portray the improved performance of the IDEFS-DNN technique, wide-ranging experiments take place on benchmark datasets and the results are inspected under varying aspects.

2. Existing Works on IDS for WSN

Sajjad et al. [11] proposed an IDS-based estimation of the trust of neighboring nodes. In the presented method, all the nodes observe the trust level of their neighboring node. According to this trust value, neighboring nodes might be stated as malicious, trustworthy, or risky. A trustworthy node is suggested to the forward engine for packet transmitting purposes. Sun et al. [12] establish a 3-level detection method by using the cooperation of the BS, ordinary nodes, and detection nodes considering their distinct abilities, adapt the V-detector procedure by adapting the detector generation rules and optimizing the detector and employ the PCA method for reducing the detection feature. The mature detector set and memory detector sets are utilized for detecting intrusions.

Mehmood et al. [13] suggest a knowledge-based context-aware method to handle the intrusion produced by malicious nodes. The activities are considered and the CH is acknowledged for blocking maliciously recurrent events produced. Zhang et al. [14] projected an enhanced cluster-based method for IDS with GA, for making an enhanced agent election procedure and adaptive IDS that is based on resource status and prevailing network conditions. Now, the experimental result has shown that with this network design the network stability and efficiency have improved widely. Ioannou and Vassiliou [15] presented mIDS that detects and monitors attacks with a statistical analysis tool-based BLR method. mIDS take input local as node parameter for benign and malicious behaviors and derive a usual behaviors method that identifies anomalies within the limited nodes.

DOI: https://doi.org/10.54216/JISIoT.000204

3. Materials and Methods

This study has presented a novel IDEFS-DNN model for intrusion detection in WSN. The proposed IDEFS-DNN model aims to select the optimum set of features and classify the intrusions in the network. In addition, the IDEFS-DNN technique involves the design of the IDEFS technique to choose a subset of optimum features. Moreover, the chosen features are fed into the DNN model for classification purposes. The usage of the IDEFS technique helps to reduce the complexity and increase the classifier outcome. Fig. 1 showcases the overall process of the IDEFS-DNN model.

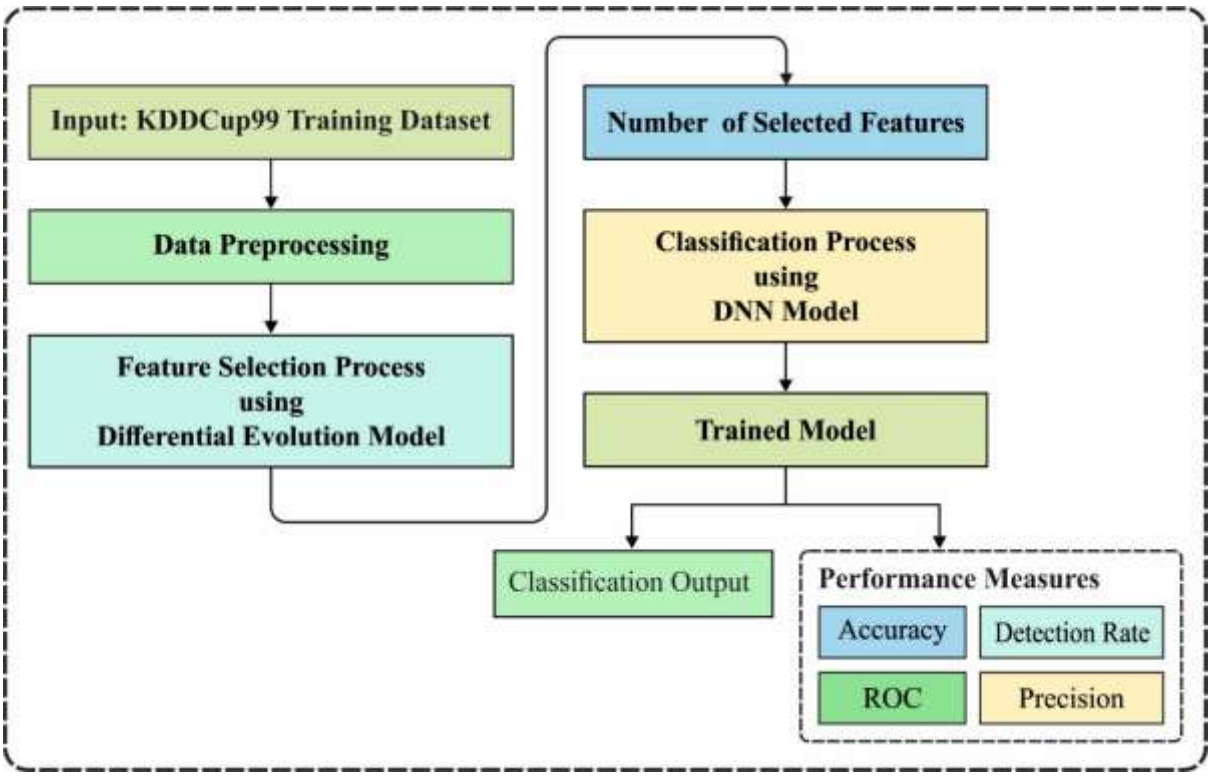

Figure 1: Overall process of IDEFS-DNN model

3.1 Algorithmic design of DE-based Feature Selection

Assume, the M amount of accessible features for a provided classifiers are represented as $F_1, \ldots, F_M$. Let, $\mathcal{A} = \{F_i : i = 1; M\}$. Then, The FS problems are specified in the following: Discover the suitable subsets of feature $\mathcal{A}' subseteq \mathcal{A}$ thus the classifiers trained by this feature must have few metrics. In this study, they enhance the F- measure values which are the integration of precision and recall. When the overall amount of features is F, the length of the chromosome is F. E.g., the encoder of a certain chromosome is denoted in Fig 1. Now, $\mathcal{F} = 12$ (viz, overall of twelve distinct features are accessible). The chromosome signifies the usage of seven features to construct a classifier (1^{st}, 2^{nd}, 4^{th}, 7^{th}, 10^{th}, $11^{th,}$ and 12^{th} features). The entries of all the chromosomes are arbitrarily initiated to 0 or 1. Now, the i^{th} location of the chromosome is 0 which characterizes that the i^{th} feature doesn't contribute to creating the classifier. Otherwise, it is 1, the i^{th} feature contributes to creating the classifier.

When the population size, each P amount of chromosomes of this population is initiated.

Fitness Computation

For the fitness calculation, the succeeding phases are implemented.

1. Assume, there are N amount of features present in a particular chromosome (viz., there are overall N amounts of ones in that chromosome).

2. Create a classifier with these N features.

DOI: https://doi.org/10.54216/JISIoT.000204

3. Now, the trained information is separated into three portions. The abovementioned classifiers are trained by the 2/3 portions of the trained information with the feature encoding in that chromosome and estimated by the residual 1/3 portion.

4. Here, the total precision, recall, and F- measure values of these classifiers for the 1/3 trained information are estimated.

5. Steps 3 and 4 are repetitive 3 times to execute 3- fold cross-validation. The average F- measure values are employed as the objective function. Therefore, the objective function corresponding to a certain chromosome is $f_1 = F - measure_{avg}$. The aim is to increase this objective function with the searchability of DE.

Mutation

For all the target vectors $x_{i,G}; i = 1,2,3, \dots, NP$, a donor vector or mutant vector is produced based on

$$v_{i,G+1} = x_{r1,G} + F(x_{r2,G} - x_{r3,G}), \quad (1)$$

whereas $r1, r2, and\ r3$ represent the haphazard indices and belong to $\{1, 2, \dots, NP\}$. They are integer values, equally distinct, and $F > 0$. Also, arbitrarily selected integers $r1, r2$ and $r3$ are selected to be distinct from the running index i, therefore NP should be equal/greater to 4 for allowing these conditions [17]. F denotes a real and constant factor 0.5 $[0,1]$ that control the extension of the differential variations $(x_{r2,G} - x_{r3,G})$.

Crossover

to raise the variety of the perturbed parameters, the crossover is presented. This is familiar as reintegration. Lastly, the trial vector:

$$u_{i,G+1} = (u_{1i,G+1}, u_{2i,G+1}, \dots, u_{Di,G+1}) \quad (2)$$

is made, whereas

$$u_{j,i,G+1} = v_{j,i,G+1} if\ (randb\ (j) \leq CR)\ \ or\ \ j = rnbr(i) \quad (3)$$

$$= x_{j,i,G} if\ (randb\ (j) > CR)\ and\ j \neq rnbr(i) \quad (4)$$

for $j = 1,2, \dots, D$,

In Eq. (3), $randb\ (j)$ denotes the jth assessment of an arbitrary value generator with results belonging to zero and one. CR denotes the crossover constant belonging to zero and one that should be defined by the user. Now, the values of CR are 0.5. (i) is an arbitrarily selected index x belonging to $\{1, 2, \dots, D\}$ that guarantees which $u_{i,G+1}$ get at minimum one variable from $v_{i,G+1}$.

Selection

For deciding it must turn out to be a member of generation $G + 1$, the trial vectors $u_{i,G+1}$ is related to the targeted vector $x_{i,G}$ with the greedy principle. When the vector $u_{i,G+1}$ produces a small cost function value than $x_{i,G}$, next $x_{i,G+1}$ is fixed to $u_{i,G+1}$, or else, the old values $x_{i,G}$ are maintained.

End criteria

In this method, the process of selection, mutation, crossover (or, reintegration), and fitness calculation is implemented for a maximal amount of generations. The optimal string to the final generation offers the optimal subsets of features. Now the optimal string comprises a collection of features. These optimal subsets of features for NER problems for a certain language.

DOI: https://doi.org/10.54216/JISIoT.000204

3.2 The process involved in DNN-based Classification

For the classification process, the DNN model is applied to determine the presence of intrusions. ANN is a computational intelligence approach inspired by the network of biological neurons to resolve forecast issues, NLP and drug identification, etc. The DNN has been NN with a fixed level of effort, and NN with several layers. The DNN utilizes a difficult mathematical system to process the data from the difficult approach. DNN with several layers joins the feature extraction as well as classification procedure to signal the learning body and creates the decision-making purpose. Usually, the DNN involves of input layer to the raw descriptor X_l, L hidden layer, and resultant layer to the data forecast. Fig. 2 showcases the framework of DNN.

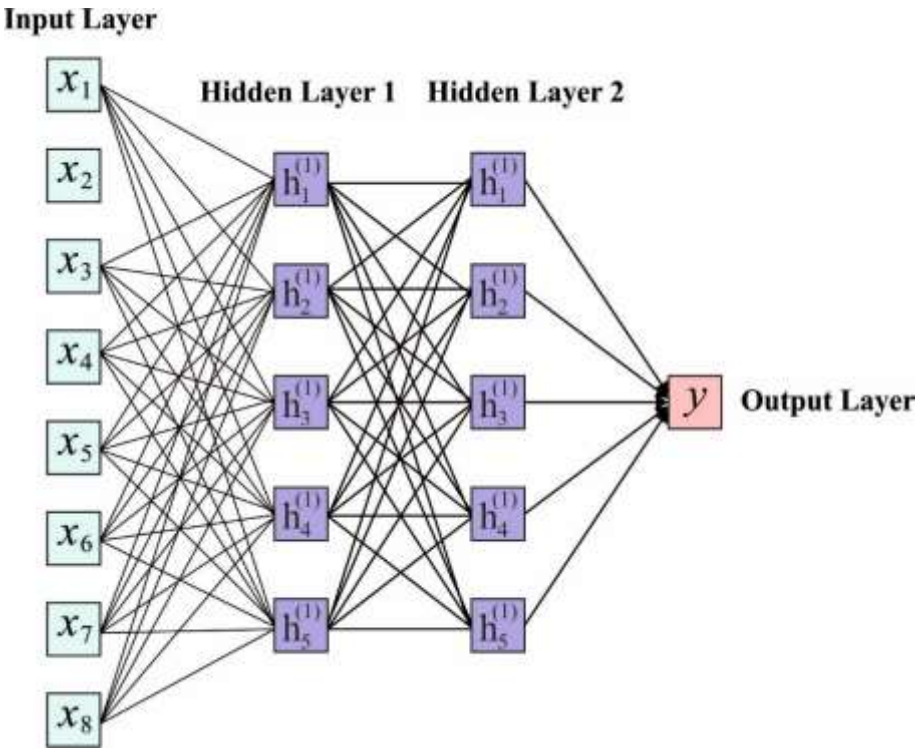

Figure 2: DNN structure

The DNN was established with utilize of the TensorFlow structure, the tf.contrib.learn.DNNClassifier DL library in Google, under the Python programming language. At this time, neither convention technique generates a better NN with an appropriate amount of layers and neurons count to all the layers. Therefore, the DNN has been generated by carrying out an extensive group of trials. In all trials, the manual structure of DNN occurs by altering the subsequent parameters: the number of hidden layers, activation function, amount of learning step, and, to all hidden layers, the number of neurons that compose the layer. Following this complex manual stage, an optimum classification efficiency was obtained with DNN collected from 7 hidden layers. The DNN Classifier class utilized now creates every neuron layer, utilizing the ReLU activation functions. In Eq. (5) of the purpose, it can be obvious that DNN has easier and effectual [18]. The resultant layer is dependent upon the softmax functions, and the cost function has been cross-entropy. The rectifier was activation purpose demonstrated in Eq. (5):

$$f(x) = x^{+} = \max(0, x) \tag{5}$$

where x refers to the input to neurons. The unit employing the rectifier is called ReLU. The smooth estimate to rectifier was analytic functions:

$$f(x) = \ln[1 + \exp(x) \tag{6}$$

DOI: https://doi.org/10.54216/JISIoT.000204

That is named soft plus functions. In the forecast technique, a novel demonstration of the raw descriptor is removed in the hidden layers as:

$$X_{t+1} = H(W_l X_l + B_l), \qquad l = 1, \dots, L \tag{7}$$

where W_l and B_l signify the weight matrix as well as a bias to $the\ l^{\text{th}}$ hidden layer, and H refers to the connected activation functions that are selected that ReLu. The pseudocode describing the steps of DNN is provided under:

- loading the trained as well as a tested group of IDS dataset
- create the classifier employing tf.contrib.learn.DNNClassifier Google library dependent upon the selected manual configuration, for instance, the number of hidden layers, activation function, amount of learning steps, and, to all hidden layers, neuron number to make the layer;
- appropriate the model employing the classifier. fit purpose;
- calculate the accuracy of DNN from the trained set employing a classifier. evaluate function;
- determine the forecast of DNN under the tested set employing the classifier. predict purpose;
- evaluate the classification outcomes of DNN from the tested set utilizing the confusion matrix;
- Confirm the classification outcomes of DNN on the complete IDS dataset.

4. Performance Validation

The performance validation of the IDEFS-DNN technique takes place using the benchmark KDD Cup 1999 dataset. It includes 125973 samples with 41 features and two classes.

Table 1 provides the best cost analysis and features chosen by the DE-FS with other techniques. Fig. 3 shows the best cost analysis of the DE-FS with existing FS techniques. The results have shown that the PSO-FS technique has obtained worse performance with the maximum best cost of 0.009139. At the same time, the BGOA-FS and FS-FS techniques have obtained moderate best costs of 0.006530 and 0.008150 respectively. However, the DE-FS technique has surpassed the other FS models with the least good cost of 0.002154.

Table 1: Result Analysis of Existing Feature Selection with Proposed DE Feature Selection Method for Applied Dataset

Methods	Best Cost	Selected Features
DE-FS	0.002154	1,2,4,5, 14,16,18,20,22,34,36,37,39
BGOA-FS	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
GA-FS	0.008150	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
PSO-FS	0.009139	3,5,8,13,18,19,20,21,22,23,25,26,27,28,30,32,33,34,36,37,38,40

DOI: https://doi.org/10.54216/JISIoT.000204

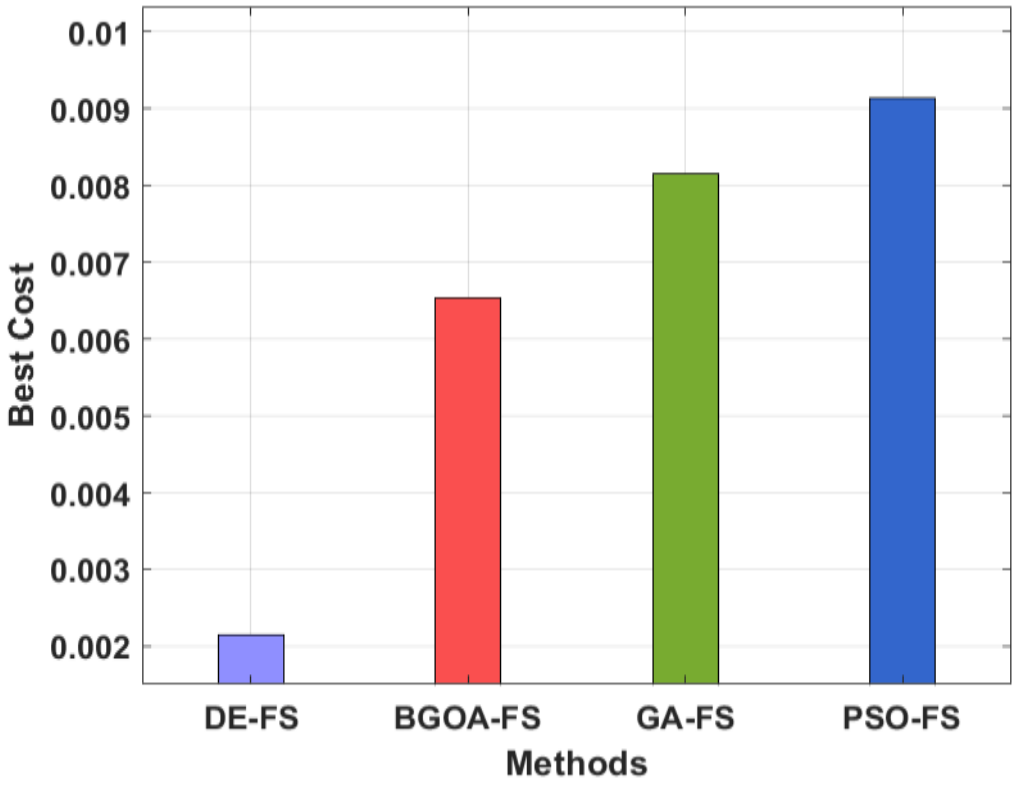

Figure 3: FS analysis of DE-FS model

Accuracy analysis of the IDEFS-DNN technique with existing techniques is examined in Table 2 and Fig. 4. The figure depicted that the DT model has showcased the last performance with the accuracy of 88.68% and 91.04% under with and without FS respectively. At the same time, the SVM+GBT+DT+RF+LR model has obtained a slightly increased accuracy of 89.95% and 92.03% under with and without FS respectively. In line with this, the LR technique has accomplished moderate accuracy of 90.37% and 90.13% under with and without FS respectively. Moreover, the RF model has gained reasonable outcomes with the accuracy of 90.69% and 89.39% under with and without FS respectively. However, the proposed IDEFS-DNN technique has outperformed the existing techniques with a maximum accuracy of 97.89% and 99.87% under with and without FS respectively.

Table 2: Result Analysis of Proposed with Existing Methods in terms of Accuracy

Methods	Accuracy (%)		Precision (%)		Detection Rate (%)	
	None	With FS	None	With FS	None	With FS
IDEFS-DNN Model	97.89	99.87	97.84	99.06	99.81	99.94
Decision Tree	88.68	91.04	90.00	91.90	98.85	99.03
Random Forest	90.69	89.39	90.90	90.70	99.33	99.42
Logistic Regression	90.37	90.13	91.50	91.30	95.78	99.29
SVM+GBT+DT+RF+LR	89.95	92.03	91.20	92.80	99.34	99.38

Precision analysis of the IDEFS-DNN manner with presented approaches is studied in Fig. 5. The figure outperformed that the DT model has showcased last performance with the precision of 90% and 91.9% under with and without FS respectively. Also, the RF model has obtained a somewhat higher precision of 90.90% and 90.70% under with and without FS respectively. In line with this, the SVM+GBT+DT+RF+LR technique has accomplished moderate precision of 91.20% and 92.80% under

DOI: https://doi.org/10.54216/JISIoT.000204

with and without FS correspondingly. Moreover, the LR model has gained reasonable outcomes with the precision of 91.50% and 91.30% under with and without FS correspondingly. But, the projected IDEFS-DNN technique has demonstrated the existing techniques with the maximal precision of 97.84% and 99.06% under with and without FS correspondingly.

A DR analysis of the IDEFS-DNN technique with existing techniques is examined in Fig. 6. The figure showcased that the LR model has exhibited the least performance with the DR of 95.78% and 99.29% under with and without FS correspondingly. At the same time, the DT model has obtained a slightly increased DR of 98.85% and 99.03% under with and without FS correspondingly. Along with that, the RF manner has accomplished moderate DR of 99.33% and 99.42% under with and without FS respectively. Moreover, the SVM+GBT+DT+RF+LR manner has reached a reasonable outcome with the DR of 99.34% and 99.38% under with and without FS respectively. Finally, the proposed IDEFS-DNN technique has outperformed the existing techniques with the maximum DR of 99.81% and 99.94% under with and without FS correspondingly.

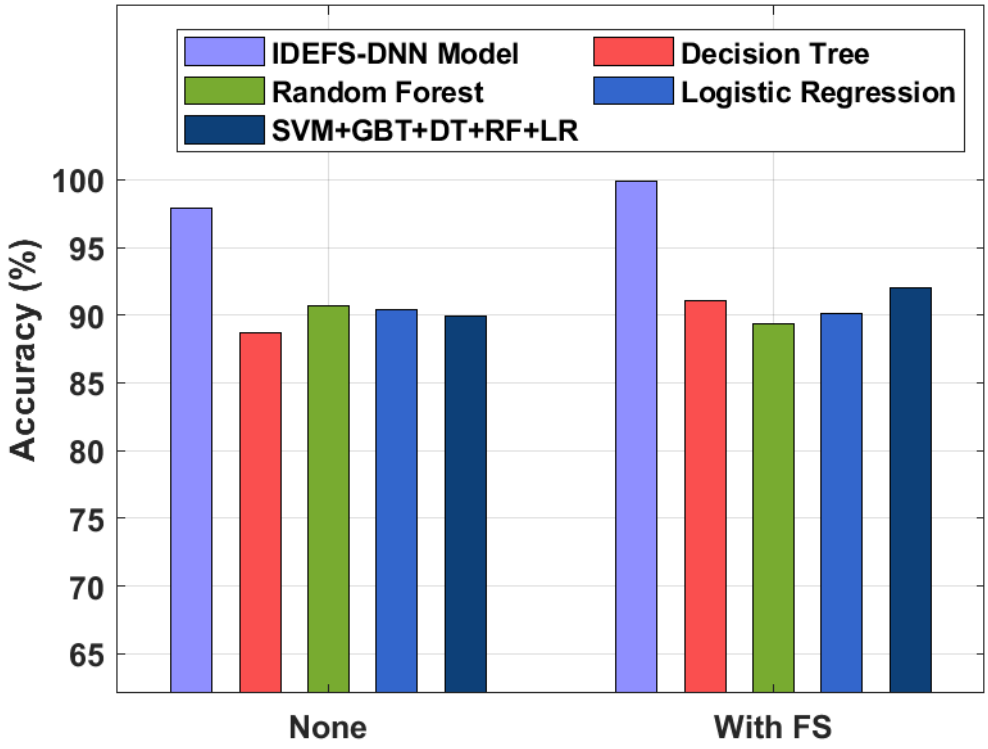

Figure 4: Accuracy analysis of IDEFS-DNN technique with existing manners

DOI: https://doi.org/10.54216/JISIoT.000204

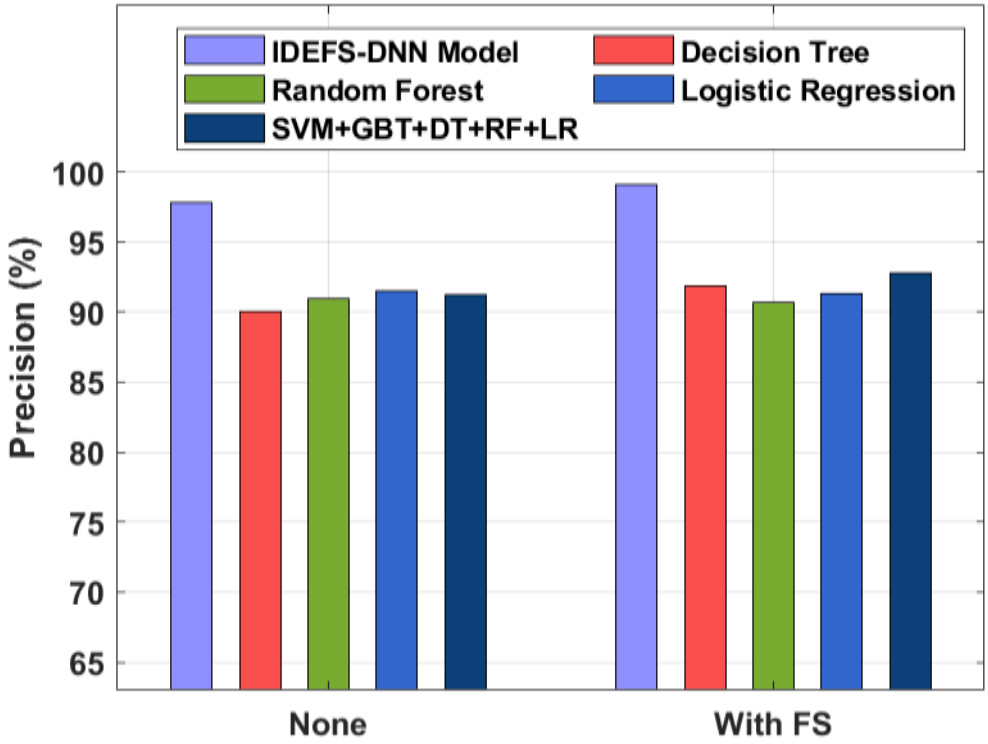

Figure 5: Precision analysis of IDEFS-DNN technique with existing manners

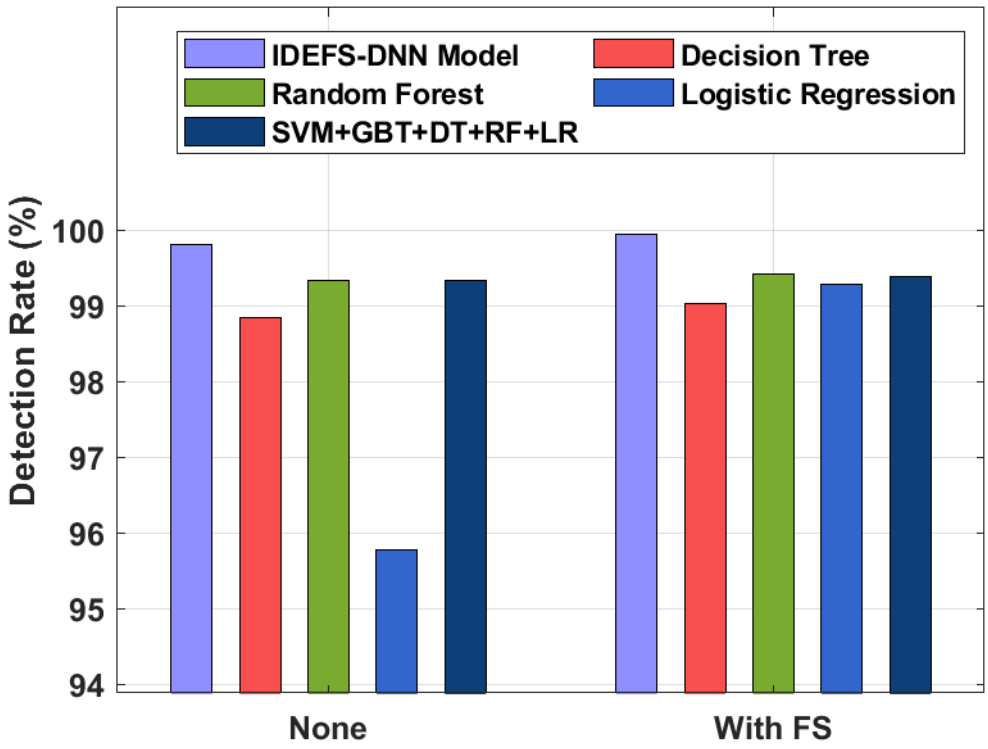

Figure 6: Detection rate analysis of IDEFS-DNN technique with existing manners

DOI: https://doi.org/10.54216/JISIoT.000204

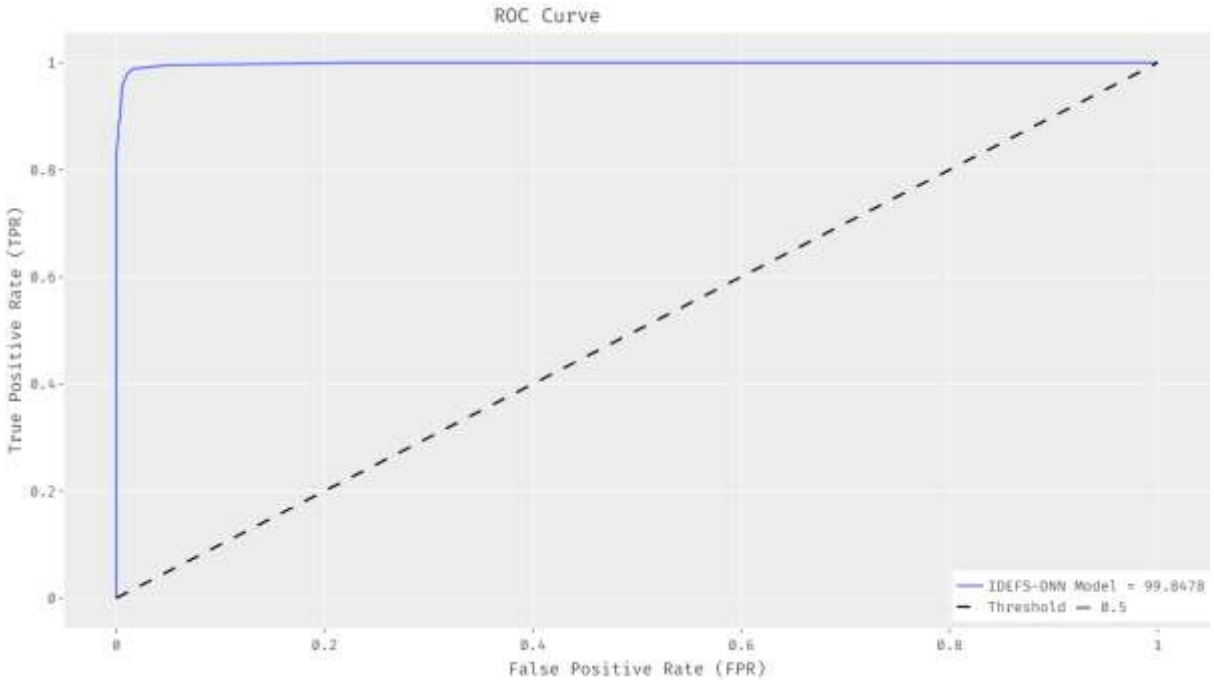

Figure 7: ROC analysis of IDEFS-DNN model

Fig. 7 depicts the ROC analysis of the IDEFS-DNN technique. The figure outperformed the IDEFS-DNN technique has resulted in an improved ROC of 99.8478.

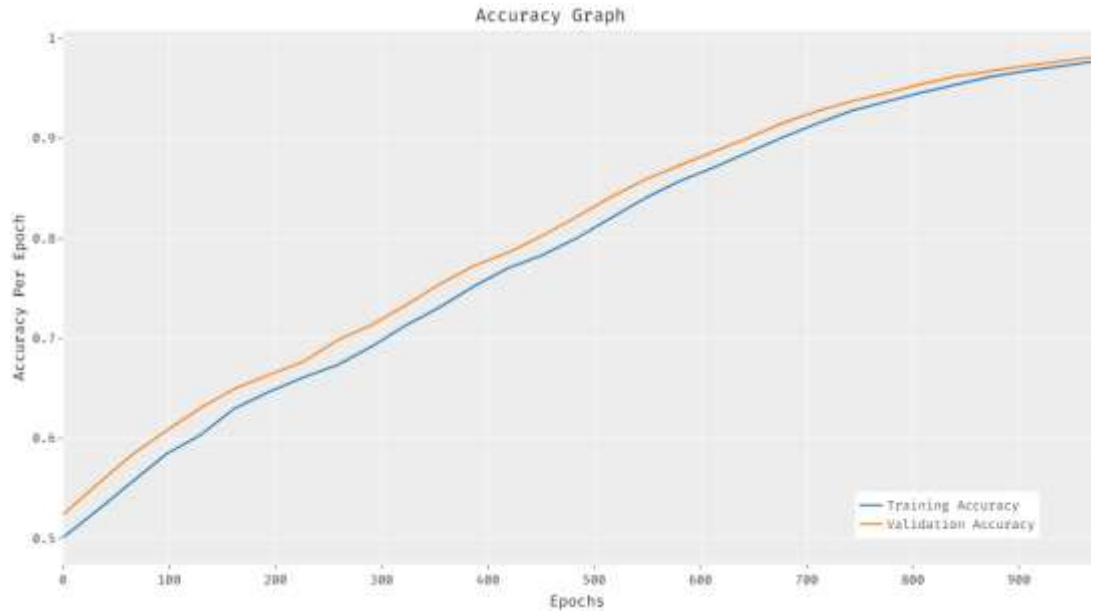

Figure 8: Accuracy analysis of IDEFS-DNN model

The accuracy graph analysis of the IDEFS-DNN system is demonstrated in Fig. 8. The figure stated that the accuracy values become enhanced with growth in epoch count. It is also noted that the validation accuracy has appeared that superior to the training accuracy.

DOI: https://doi.org/10.54216/JISIoT.000204

The loss graph analysis of the IDEFS-DNN manner is shown in Fig. 9. The figure inferred that the loss values are initiated to lower with a rise in epoch count. It is also stated that the validation loss is considerably minimal than the training loss.

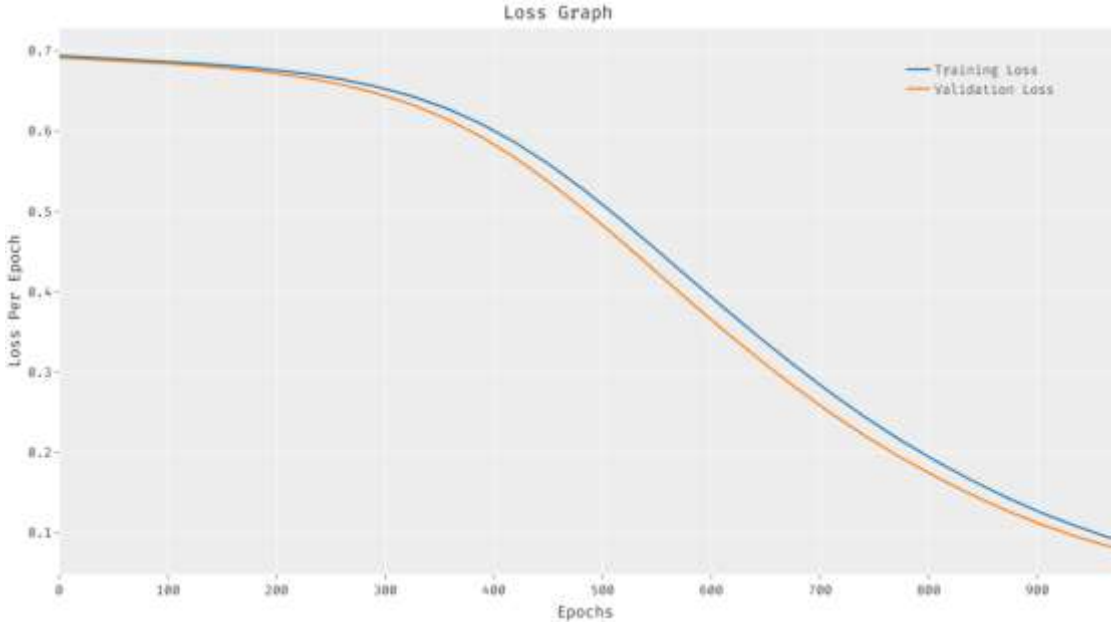

Figure 9: Loss analysis of IDEFS-DNN model

5. Conclusion

This study has presented a novel IDEFS-DNN technique for intrusion detection in WSN. The proposed IDEFS-DNN technique aims to select the optimum set of features and classify the intrusions in the network. In addition, the IDEFS-DNN technique involves the design of the IDEFS technique to choose a subset of optimum features. Moreover, the chosen features are fed into the DNN approach for classification purposes. The usage of the IDEFS technique helps to reduce the complexity and increase the classifier outcome. To portray the improved performance of the IDEFS-DNN technique, wide-ranging experiments take place on benchmark datasets and the results are inspected under varying aspects. The simulation results ensured the enhanced intrusion detection performance of the IDEFS-DNN technique over the other IDS models. In the future, hybrid DL models can be utilized in place of DNN to enhance the detection rate.

References

[1] Almomani, I., Al-Kasasbeh, B. and Al-Akhras, M., 2016. WSN-DS: A dataset for intrusion detection systems in wireless sensor networks. Journal of Sensors, 2016.

[2] Can, O. and Sahingoz, O.K., 2015, May. A survey of intrusion detection systems in wireless sensor networks. In 2015 6th international conference on modeling, simulation, and applied optimization (ICMSAO) (pp. 1-6). IEEE.

[3] Ioannou, C., Vassiliou, V. and Sergiou, C., 2017, May. An intrusion detection system for wireless sensor networks. In 2017 24th International Conference on Telecommunications (ICT) (pp. 1-5). IEEE.

[4] McDermott, C.D. and Petrovski, A., 2017. Investigation of computational intelligence techniques for intrusion detection in wireless sensor networks. International journal of computer networks and communications, 9(4).

[5] Abdullah, M.A., Alsolami, B.M., Alyahya, H.M. and Alotibi, M.H., 2018. Retracted: Intrusion detection of DoS attacks in WSNs using classification techniuqes. Journal of fundamental and Applied Sciences, 10(4S), pp.298-303.

DOI: https://doi.org/10.54216/JISIoT.000204

[6] Chandre, P.R., Mahalle, P.N. and Shinde, G.R., 2018, November. Machine learning based novel approach for intrusion detection and prevention system: A tool based verification. In 2018 IEEE Global Conference on Wireless Computing and Networking (GCWCN) (pp. 135-140). IEEE.

[7] Gara, F., Saad, L.B. and Ayed, R.B., 2017, June. An intrusion detection system for selective forwarding attack in IPv6-based mobile WSNs. In 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC) (pp. 276-281). IEEE.

[8] Sandhya, R. and Sengottaiyan, N., 2017. Dynamic ch selection and intrusion detection in wsn using reinforced weighted approximation based adaptive seech: An optimized routing framework. International Journal of Applied Engineering Research, 12(20), pp.9315-9326.

[9] Mekelleche, F. and OuldBouamam, B., 2018, April. Monitoring of Wireless Sensor Networks: Analysis of Intrusion Detection Systems. In 2018 5th International Conference on Control, Decision and Information Technologies (CoDIT) (pp. 421-426). IEEE.

[10] Sahoo, K.C. and Pati, U.C., 2017, May. IoT based intrusion detection system using PIR sensor. In 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) (pp. 1641-1645). IEEE.

[11] Sajjad, S.M., Bouk, S.H. and Yousaf, M., 2015. Neighbor node trust based intrusion detection system for WSN. Procedia Computer Science, 63, pp.183-188.

[12] Sun, Z., Xu, Y., Liang, G. and Zhou, Z., 2017. An intrusion detection model for wireless sensor networks with an improved V-detector algorithm. IEEE sensors journal, 18(5), pp.1971-1984.

[13] Mehmood, A., Khanan, A., Umar, M.M., Abdullah, S., Ariffin, K.A.Z. and Song, H., 2017. Secure knowledge and cluster-based intrusion detection mechanism for smart wireless sensor networks. IEEE Access, 6, pp.5688-5694.

[14] Zhang, Z., Zhu, H., Luo, S., Xin, Y. and Liu, X., 2017. Intrusion detection based on state context and hierarchical trust in wireless sensor networks. IEEE Access, 5, pp.12088-12102.

[15] Ioannou, C. and Vassiliou, V., 2018, October. An intrusion detection system for constrained WSN and IoT nodes based on binary logistic regression. In Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (pp. 259-263).

[16] Sikdar, U.K., Ekbal, A. and Saha, S., 2012, December. Differential evolution based feature selection and classifier ensemble for named entity recognition. In Proceedings of COLING 2012 (pp. 2475-2490).

[17] Hossen, T., Nair, A.S., Chinnathambi, R.A. and Ranganathan, P., 2018, September. Residential load forecasting using deep neural networks (DNN). In 2018 North American Power Symposium (NAPS) (pp. 1-5). IEEE.

DOI: https://doi.org/10.54216/JISIoT.000204