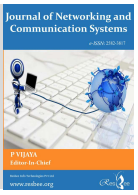
Vol.7 No.3 July 2024

Journal of Networking and Communication Systems

Received 9 July, Revised 9 August, Accepted 2 September



JS-DKN: Jensen Shannon with Deep Kronecker Networks for Feature Fusion in IoT Intrusion Detection

Shanoor Afreen

Department of Computer Science
University of Texas at Arlington, United States
shanoorafreen.2311@gmail.com

Abstract: Intrusion detection is considered the process of examining or monitoring traffic in the network, user behavior, and system function, to recognize the unofficial anomalous or access actions that could indicate the cyberattacks or security breaches. In intrusion detection, unifying multiple features causes redundancy, and irreverent features are added and causing a waste of computational resources. Thus, this research presented a new technique named Jensen Shannon Deep Kronecker Networks with Deep Stacked Autoencoders (JS-DKN +DSA) for feature fusion in IoT intrusion detection. At first, the simulation of IoT is performed, and thereafter, the input network traffic data is applied to data normalization. Furthermore, the z-score is utilized for data normalization to improve the quality of data. Thereafter, feature fusion is performed by the Jensen Shannon with Deep Kronecker Networks (JS-DKN), in which the JS is employed to find the relevant features, and the DKN is applied to find the fusion coefficient. Lastly, intrusion detection in IoT is effectuated by the Deep Stacked Autoencoder (DSA) and produces output as attack or normal. In addition, the presented JS-DKN +DSA approach recorded the highest accuracy of 90.876%, True Negative Rate (TNR) of 88.654%, and a True Positive Rate (TPR) of 92.766 %.

Keywords: IoT, intrusion detection, deep learning, feature fusion, Jensen Shannon similarity.

Nomenclature

Abbreviation	Expansion
JS-DKN	Jensen Shannon Deep Kronecker Networks
DSA	Deep Stacked Autoencoders
TNR	True Negative Rate
TPR	True Positive Rate
IoT	Internet of Things
IDS	Intrusion Detection Systems
IT	Information Technology
DL	Deep Learning
ML	Machine Learning
DNN	Deep Neural Networks
NSBPSO	Neighborhood Search-Based Particle Swarm Optimization
IMIDS	Intelligent Intrusion Detection System
XGB-RF	eXtreme Gradient Boosting Random Forest
AEs	Autoencoders
DoS	Denial of Service

1. Introduction

The IoT is considered an advanced framework standard envisioned as a global network of devices and machines capable of communication with each other [10]. IoT refers to the smart network because it is applied to introducing the protocols to connect things to the internet [2] [13]. The IoT is comprised of a network of physical smart objects with software, communication and computing components, and sensors to collect and transform information with other related devices around the world [10]. Based on the reports of analytics, there are more than 13.8 billion IoT devices utilized worldwide, and the count is expected to grow to 30.9 billion by 2025 [10]. The IoT devices are applied in multiple applications, such as smart homes, security, supply-chain management, industrial production, Blockchain monitoring, and transportation. A large information is generated by IoT devices. However, the nature of the data maintains a common

https://doi.org/10.46253/jnacs.v7i3.a4

pattern [1]. Moreover, IoT networks are subject to an array of malicious attacks because IoT devices are accessed from anywhere over an insecure network namely the Internet [12]. Yet, the significant difficulty acquired by the IoT system is security, and among all the kinds of security mechanisms, intrusion detection is referred to as a vital security mechanism [2]. A crucial challenge in introducing efficient IDS for IoT applications is managing huge volumes of information while minimizing energy consumption and conserving information privacy [11].

An intrusion refers to the arrangement of unpredictable activities network confidentiality, integrity, globally or locally, harming and/or availability. The IDS is considered an efficient security measure for identifying malicious actions or cyber-attacks on computer systems [10]. IDS are usually utilized for improving the security measures in an IT infrastructure [3]. Furthermore, the IDS are classified regarding the observed information source as either network-based or host-based. The host-based IDS examines information emerging from the logs namely application, database logs, and operating system of the host systems. It can monitor the behavior of objects of high importance, like sensitive files or programs, and accurately detect intrusions or abnormal actions [10]. The Network IDS (NIDS) is considered a capable solution for detecting intrusions based on malicious behaviors in IoT networks. In the IoT, the NIDS is enabled in the network layer and plays as a spine to link numerous IoT devices [2]. Several types of DL approaches are utilized in distinct modules of the whole procedure [14]. DL is a sub-field of ML that contains numerous hidden layers making it more appropriate to work with issues with massive data [10]. DNNs are efficiently used for analyzing large volumes of data, establishing patterns, and relations among them, and classifying the information based on distinct properties [14]. In IoT security, the application of DNN is a considered promising solution to network data anomaly-based intrusion detection [1].

This article aims to develop a novel approach of JS-DKN +DSA for feature fusion in IoT intrusion detection. Primarily, the IoT is stimulated, and the input network traffic data is forwarded to data normalization. Then the data normalization is done by the z-score to enhance the data quality. Consequently, the JS-DKN is employed for feature fusion, here, the relevant features are found regarding the JS and the features fusion is done based on the fusion coefficient attained using DKN. At last, intrusion detection in the IoT is done by DSA.

The significant contribution of this article is illustrated below,

- ***Proposed JS-DKN +DSA for feature fusion in IoT intrusion detection*:** The JS-DKN technique is introduced for feature fusion, where, JS is used for attaining the relevant features and the DKN is applied for the feature correlation in the fusion process, and the introduced DSA method is established for IoT intrusion detection.

The remnant sections of this research work are arranged as follows, the prevailing methodologies for the detection of intrusion in IoT are demonstrated in Section 2, Section 3 depicts the system model for IoT, the introduced method is elucidated in Section 4, the experimental outcome is illustrated in Section 5 and the conclusion is illustrated in the section 6.

2. Motivation

IoT is considered the most significant concept in numerous aspects of our current life in nowadays. Moreover, the IoT system has caused a substantial increase in data traffic and subsequently high dimensionality. Yet, security issues are considered as the most serious problems faced during the utilization of the IoT and cyber threats become a serious issue, particularly for the IoT servers. Therefore, the JS-DKN +DSA is presented for feature fusion in IoT intrusion detection.

2.1 Literature Review

Awajan, A., [1] presented the Deep Learning-based Intrusion Detection (DL-based ID) for the detection of attacks in IoT devices. This technique was robust and scalable, however, this technique did not investigate more attacks on IoT devices to ensure better security and strengthen the approach in the IoT networks. Baniasadi, S., *et.al* [2] developed the NSBPSO algorithm for the detection of intrusion in IoT Systems. This framework had better accuracy and performance and reduced the computational cost. However, this method did not consider a large number of the dataset and failed to explore the time complexity. Le, K.H., *et.al* [3] established the IMIDS for detecting intrusion in IoT. This approach provided a classification of multiple cyber threats and enhanced performance. Yet, this approach required a long time for the training process. Faysal, J.A., *et.al* [4] introduced the XGB-RF classifier for IoT intrusion detection. This method could detect botnet attacks effectively and security features were improved in the IoT schemes. However, this technique did not explore the time consumption in intrusion detection while maintaining higher accuracy. Albulayhi, K., *et.al* [5] devised the feature selection approach with a bagging classifier for intrusion detection in IoT. This method minimized the dimensionality, conserved resources, and reduced training time consumption. Nevertheless, this approach failed to investigate the deployment of the IoT

gateway device for the process of delivery of classification and detection services as opposed to numerous intrusions and cyber-attacks in the IoT devices.

2.2 Challenges

The major challenges faced by the existing methodologies for intrusion detection in IoT are described as follows,

- The DL-based ID [1] approach was efficient and flexible for the detection of intrusion. Yet, this framework did not consider the development of a platform-independent approach for efficient and effective intrusion detection.
- The XGB-RF classifier [4] framework reduced security problems in the system and was robust. Nevertheless, this framework failed to analyze the performance of ML classifiers for the process of identifying unknown attacks in IoT environments.
- The feature selection approach with a bagging classifier [5] failed to investigate resource consumption and did not consider the additional evaluation parameters for improving performance.
- In recent days, the IoT has become vital in numerous aspects of our modern life. The IoT attacks often remain undetected for a long period, causing extreme service disruption and subsequent causes in economic loss and it also affects the privacy of the people.

3. System Model

In the security chain, the main and vital process is the detection of intruders. The entire intrusion process is achieved only after the detection of the attacker. Firewalls and Anti-malware software are considered the leading protectors in user terminals. Comparatively, IDS acts as the protector for the Internet servers. Here, due to the remote-control feature numerous IoT servers and IoT devices are directly exposed to the public Internet. The vulnerabilities are captured by the attackers for intruding on the IoT servers. To identify and safeguard the IoT servers from intruders, an IDS is essential [6]. The system model for the IoT network is explicated in the fig. 1.

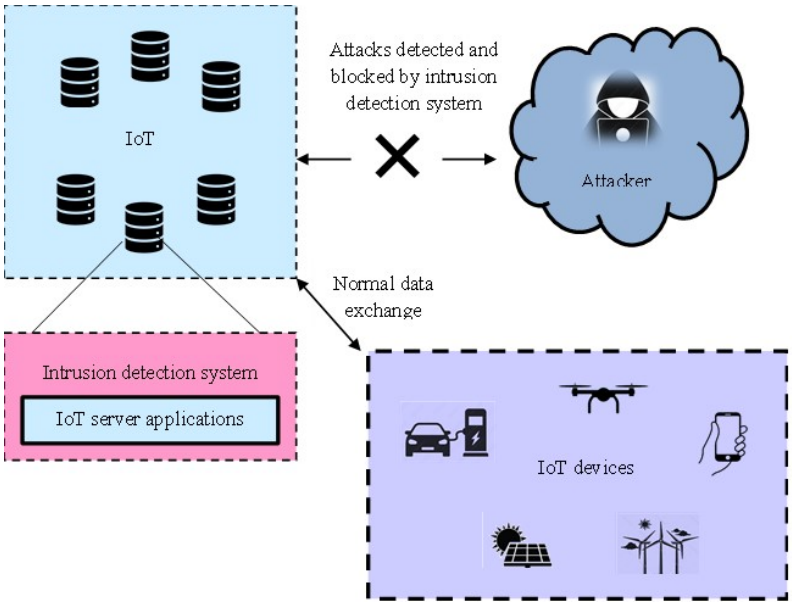


Fig. 1. *The system model for IoT network.*

4. Proposed Jensen Shannon with Deep Kronecker Networks with DSA for Feature Fusion in IoT Intrusion Detection

The process of examining and monitoring the information from the IoT to identify malicious activity and unauthorized access is defined as Intrusion detection. However, the detection of intrusion is considered a difficult task, because it produces a high false positive rate and security problems. Thus, a JS-DKN with DSA is introduced for feature fusion in IoT intrusion detection in this work. Primarily, the IoT is simulated, and then, the input network traffic data is assimilated from the dataset. Then, the data normalization is performed by the z-score [7]. Subsequently, the feature fusion is established using the JS-DKN, where the

JS is utilized for obtaining the significant features and the DKN is employed to generate the fusion coefficient. At last, the detection of the intrusion is performed by the DSA [9]. The Schematic illustration of JS-DKN with DSA [9] for feature fusion in IoT intrusion detection is shown in Fig. 2.

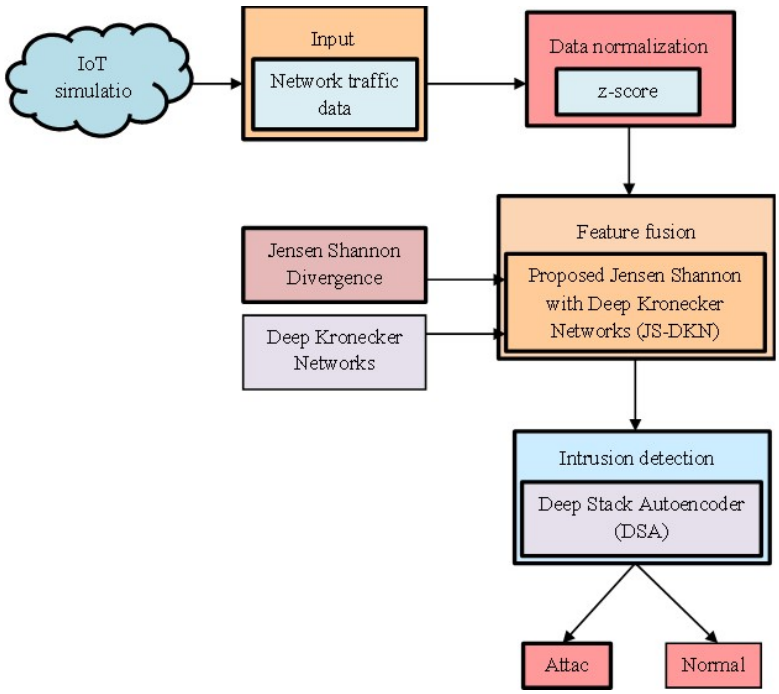


Fig. 2. *Schematic illustration of JS-DKN with DSA for feature fusion in IoT intrusion detection.*

4.1. Data Acquisition

The input IoT network traffic data is attained from the CIC IoT dataset 2023 [16], indicated as B for detecting intrusions, and the dataset is expressed as,

$$B = \{B_1, B_2, ...B_c ...., B_u\} \tag{1}$$

wherein, u indicates the overall data in the dataset, B represents the input dataset, and B_c indicates the c^{th} data that is regarded as the input.

4.2. Data Normalization

The approach used for arranging the information in the dataset to minimize redundancy and enhance data integrity is called data normalization. Data normalization is mostly used for reducing duplicate data and to make the database more efficient for updating and querying approaches. The z-score [7] is employed for the process of data normalization to improve scalability and enhance data security. The input data B_c is passed to the z-score to obtain the normalized data W_s in the data normalization process. The z-score is also referred to as the standard score, which is utilized in statistics for estimating the probability of a score existing within the normal distribution and facilitates the evaluation of two scores across several normal distributions [7]. For the z-score, every value of the input is normalized based on the below equation,

$$E(r,s) = \frac{t(r,s) - \alpha}{\beta} \tag{2}$$

where, $E(r,s)$ signifies the new value, $t(r,s)$ represents the old value, α denotes the average input value, and β denotes the column's standard deviation in the input value.

4.3. Feature Fusion

The process of integrating various features or a group of features from numerous modalities or sources to develop a more informative representation for modeling is known as feature fusion. Furthermore, feature fusion is employed to decrease the computational complexity and reduce the dimensionality. Here, the JS-DKN is utilized for performing the feature fusion, where the most relevant features are found by using the JS similarity, and fusion is established with the help of a fusion coefficient generated using DKN.

a) Sorting of features based on JS similarity

Initially, the sorting of features $Y$is done by applying the JS similarity [15]. The JS similarity is derived from the JS divergence that assesses the comparability among two probability distributions [15]. Here, the most salient features are extracted by the JS similarity that assists in the detection process. Furthermore, the JS similarity had robustness and is it expressed as,

$$\delta\left(Y,Z\right)=0.5*\left(\sum Y_b*\log\left(2*Y_b\left|Y_b+Z_b\right.\right)+\sum Z_b\log\left(2*Z_b\left|Y_b+Z_b\right.\right)\right) \tag{3}$$

here, Yimplies the feature, Z signifies the target, b indicates the b^{th} data point, and R_n denotes the sorted output.

b) Fusion

The parameter of the fusion coefficient ϑ is generated by using the DKN for the process of reorganizing the features, and feature fusion is formulated as follows,

$$R_d^{new}=\sum_{\substack{m=d\\ m+1=d+q}}^{h}\frac{\vartheta_m}{m}R_n \tag{4}$$

where h implies the total count of the features, q represents the quantity of the features to be selected, R_d^{new} indicates the fusion output, and ϑ indicates the coefficient of the fusion that is generated using DKN.

c) Generating ϑ using the DKN

The feature fusion is established by regarding the fusion coefficient ϑ generated by utilizing the DKN. Primarily, the sorted features are forwarded into the DKN as the input. The DKN is used in the generation of the ϑ because of its ability to establish an efficient outcome even when the sample size available is low and the fusion coefficient ϑ is formulated as shown below,

$$\vartheta=\delta\left(\chi_f,\gamma_f\right) \tag{5}$$

where, χ_f indicates the data record, and γ_f signifies the mean of χ_f associated with the class.

Architecture of DKN

The DKN is considered the kind of neural network architecture that uses the Kronecker product for modeling relationships in high-dimensional information effectively [8]. The DKN improves the efficiency of feature interaction modeling and provides robustness and scalability. The sorted feature R_n is subjected to the input in the DKN. Assumes the response ϑ utilizes the generalized linear approach as formulated below,

$$\vartheta=G\left(\vartheta\right)\exp\left\{R_n\left\langle\Re_\phi,T\right\rangle-\sigma\left(\left\langle\Re_\phi,T\right\rangle\right)\right\} \tag{6}$$

here, $G\left(.\right)$ and $\sigma\left(.\right)$ indicates defines the known univariate function, and $T\in\mathbb{R}^{a\times b}$ represents the target unknown coefficient matrix. Moreover, certain link functions $\Im i\left(.\right)$ expressed as,

$$\Im\left(j\left(\vartheta\right)\right)=\left\langle\Re_\phi,T\right\rangle \tag{7}$$

where, Tdenotes the coefficient with the rank P, and the Kronecker products decomposition with $\aleph\geq 2$ is formulated as,

$$T=\sum_{v}^{P}\eta_\aleph^v\otimes\eta_{\aleph-1}^v\otimes.....\otimes\eta_1^v \tag{8}$$

wherein, $\eta_\aleph^v=\mathbb{R}^{a\times b}$, $\kappa=1,.....\aleph, v=1,.....P$ represents the unknown matrix, and η_κ^v signifies the unknown size. Based on the Kronecker product property, they are certainly required to declare the $J=\Pi_{\kappa=1}^{\aleph}J_\kappa$ and $b=\Pi_{\kappa=1}^{\aleph}b_\kappa$ are expressed as below,

$$\eta_{\kappa'}\otimes\eta_{\kappa'-1}\otimes.....\otimes\eta_{\kappa'}=\overset{\kappa'}{\underset{\kappa=\kappa'}{\otimes}}\eta_w \tag{9}$$

For any matrix $\eta_{\kappa'},.....\eta_{\kappa'}$ with $\kappa'\geq\kappa''$.

$$T=\sum_{v=1}^{P}\otimes_{\kappa=\aleph}^{'}\eta_\kappa^{'} \tag{10}$$

Thus, the resultant of the DKN generates the feature coefficient ϑ and the framework of DKN is demonstrated in Fig. 3.

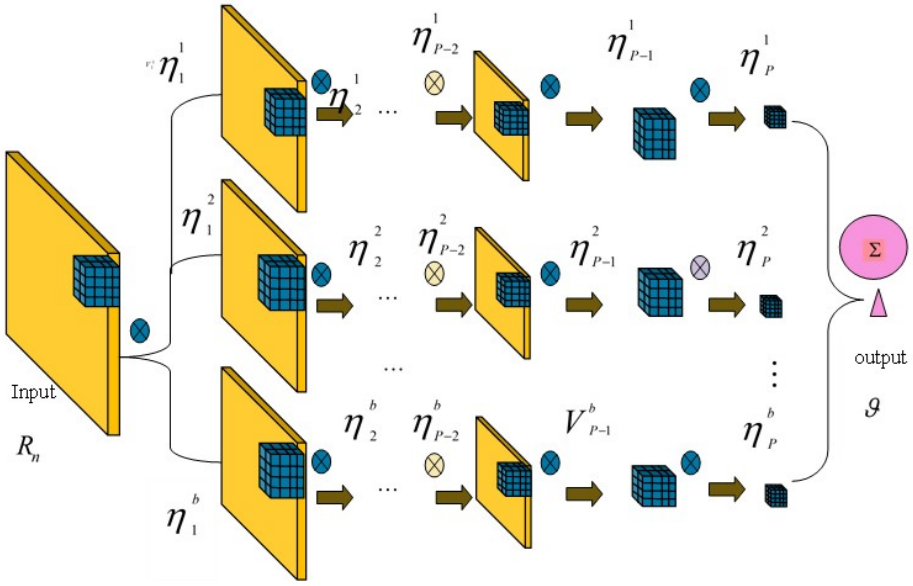


Fig. 3. *Framework of DKN*

4.4. Intrusion Detection

The process of analyzing the system or network activities to recognize illegal access or security breaches. The fused feature R_d^{new} is employed as the input for the DSA and the resultant determines whether the intrusion is an attack or normal [7]. The DSA is applied in the process of detection as it can minimize the dimensionality and reduce the computational cost.

4.4.1. Architecture of DSA

The kind of DNN is comprised of several layers of AEs and is known as the DSA [9]. Moreover, the AE is categorized into two types namely decoder and encoder. The DSA can extract the relevant features and is robust. Consequently, the encoder consists of a hidden layer, a mapping function, and an input layer, among them. Thus, the resultant of the hidden layer is embodied as,

$$A_\theta(y) = z(py + N) \tag{11}$$

here, $\mathcal{Y}$ represents the input fused feature R_d^{new}, N indicates the bias vector, p signifies the encoding weight matrix, and $z(.)$ denotes the encoder's activation function.

Furthermore, the decoder is comprised of a hidden layer, a mapping function, and an output layer, among them, and the encoder's reverse process is determined as the decoder. The resultant signals are represented as reconstructed signals, which is denoted as $\hat{y}$ and mapping relationship formulated below as,

$$\hat{y}_{\theta'}(A) = S\left(p^* A + N^*\right) \tag{12}$$

here, $S(.)$ represents the decoder's activation function, N^* indicates the bias vector, and p^* signifies the weight matrix.

The AEs generally suffer from overfitting issues during training, which can be addressed using two operations. The dropout is considered as the first operation, where the average value of numerous approach predictions so that the update weight is independent of the fixed relationship by the associated action of neurons. The second operation is by using a weighted decay in the loss function. During dropout, in the hidden layer, the neurons are frequently discarded based on the probability of forming a sub-network of the initial network. The discarded matrix H_k complies with the Bernoulli distribution and the function of the probability distribution is explained below,

$$V(l; H_k) = \begin{cases} H_k, l = 1 \\ 1 - H_k, l = 0 \end{cases} \tag{13}$$

where, l specifies the probable output and H_k denotes the discarded matrix.

Afterward, the dropout approach is presented in every hidden layer, and the mapping function of the primary neutral network is altered by the below as,

$$I = A_M(y) = U_M z\left(\sum_{l=1}^{g} pR_M + N\right) = \begin{cases} z\left(\sum_{l=1}^{g} pR_M + N\right), U_M = 1 \\ 0, U_M = 0 \end{cases} \quad (14)$$

Here, g signifies the input dimension and U_M symbolizes the M^{th} elements in the discarded matrix. At last, the outcome of the detecting intrusion is represented as I , and the general outline for the DSA is demonstrated in Fig. 4.

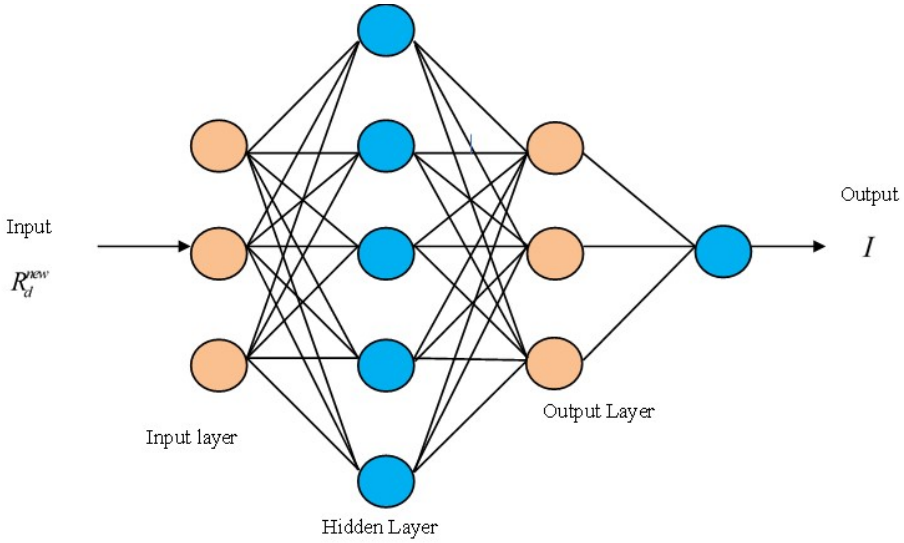


Fig. 4. *General outline for DSA*

5. Result and Discussion

The evaluation of the developed JS-DKN+ DSA for feature fusion in IoT intrusion detection is carried out concerning several metrics and the developed approach is correlated with the classical existing frameworks to determine its performance.

5.1 Experimental setup

The Python tool is applied for implementing the designed JS-DKN+ DSA to detect the intrusion.

5.2. Dataset Description

The presented JS-DKN+ DSA for intrusion detection is realized by using the CIC IoT dataset 2023 [16]. The main aim of the dataset is to foster the improvement of the security evaluating applications in the operations of IoT in real time. To achieve this, the 33 types of attacks are determined by the IoT topology from 105 devices. These types of attacks are categorized into seven types such as; Brute Force, Spoofing, Web-based, Mirai, DoS, and Recon. This dataset comprised four subdirectories correlated to the various files like PCAP, CSV, Example, and Supplementary material.

5.3. Evaluation Metrics

The JS-DKN+ DSA approach is examined for its performance because of evaluation measures such as accuracy, TPR, and TNR and it is explained below.

i)Accuracy

The proportion of accurately detected instances to the complete instances is defined as accuracy and accuracy characterized as Υ is given below,

$$\Upsilon = \frac{XB + XE}{XB + XE + HB + HE} \quad (15)$$

where HB signifies the false negative, XE represents the true negative, XB indicates the true positive, and HE denotes the false negative.

ii)TPR

TPR is also referred to as Sensitivity or Recall and it determines the ratio of the percentage of real positive occurrences that are accurately identified by the approach from the total intrusions. Further, the TPR represented as τ is given by,

$$\tau = \frac{XB}{XB + HE} \tag{16}$$

iii) TNR

TPR is the measure of the percentage of real negative occurrences that are accurately recognized by the method, and is expressed as,

$$\kappa = \frac{HE}{HB + HE} \tag{17}$$

Here, TNR is represented as κ.

5.4. Deviation Analysis

The deviation analysis for the presented JS-DKN+ DSA concerning the samples is epitomized in Fig. 5. The deviation analysis is established by considering two labels: '0' and '1'. '0' represents no intrusion and '1' indicates the intrusion. When the actual label is '1' and the predicted label is '0', a deviation of '+1' is attained. When the actual label is '0' and the predicted label is '1', the deviation is denoted by '-1'. If both labels are the same, then the deviation value is '0'.

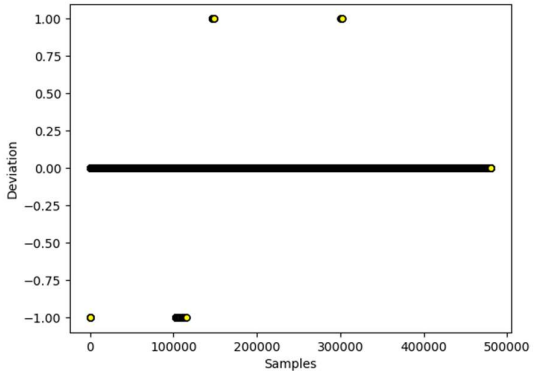


Fig. 5. *Deviation analysis of the presented JS-DKN+ DSA*

5.5. Predication Check Plot

Fig. 6. illustrates the predication check plot for the JS-DKN+ DSA. The prediction plot shows the deviation of samples classified by the JS-DKN+ DSA from the actual samples. The plot shows the classification results obtained by the JS-DKN+ DSA for both class '0' and class '1' samples.

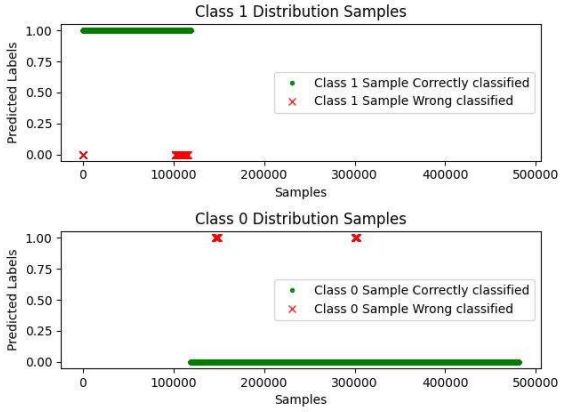


Fig. 6. *Predication check plot for the designed JS-DKN+ DSA*

5.6 Comparative Approaches

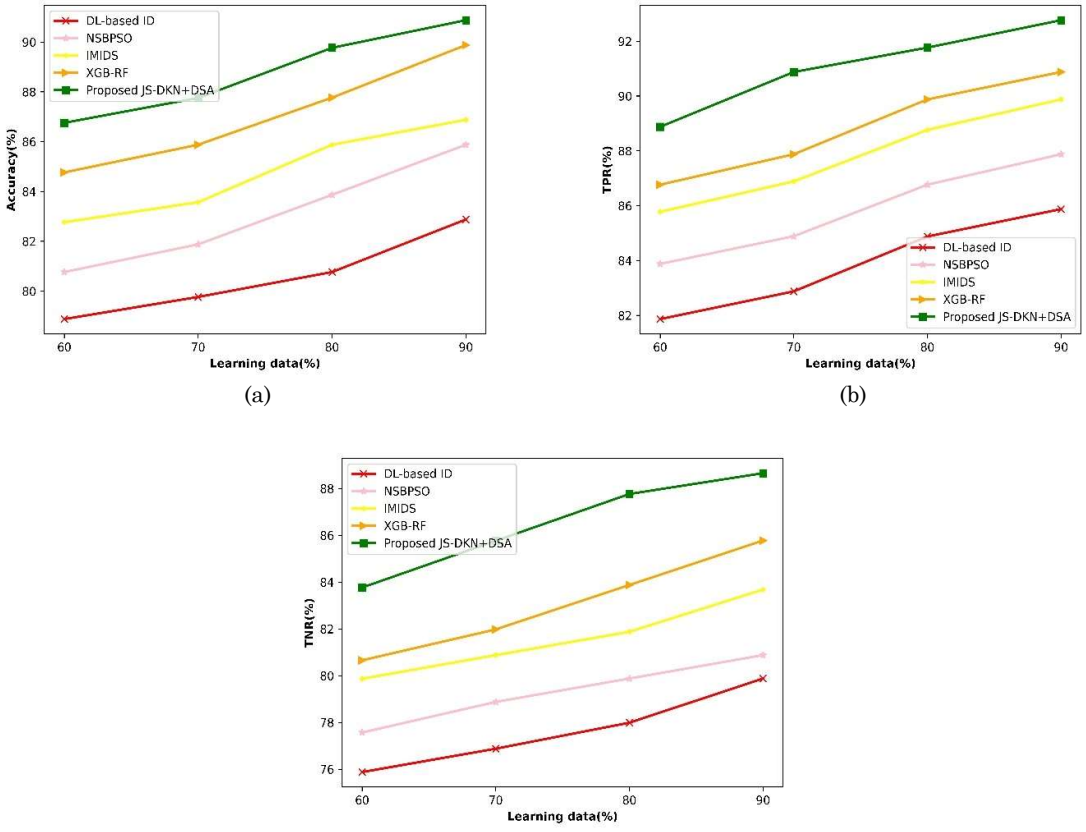
The comparative techniques utilized for the analysis of the JS-DKN+ DSA are DL-based ID [1], NSBPSO [2], IMIDS [3], and XGB-RF [4].

5.7 Comparative Assessment

The JS-DKN+ DSA method is correlated with several approaches in terms of TNR, TPR, and accuracy, which are explained as follows.

5.7.1 Comparative Investigation Concerning the Learning Data

Fig. 7 explicates the assessment of the JS-DKN+ DSA regarding the learning data. The examination of the JS-DKN+ DSA regarding the accuracy is portrayed in Fig. 7 (a). With 60% of the learning data, the JS-DKN+ DSA achieved an accuracy of 86.755%, whereas the prevailing techniques, such as DL-based ID, NSBPSO, IMIDS, and XGB-RF measured an accuracy of 78.877%, 80.766%, 82.766%, and 84.765%. The accuracy achieved by the JS-DKN+ DSA technique is 2.29% better than the DL-based ID. Fig. 7 (b) exemplified the valuation of the JS-DKN+ DSA to the TPR. The TPR recorded by the JS-DKN+ DSA is 90.877% with the learning data of 70%, and the TPR recorded by the prevailing methodologies namely DL-based ID is 82.877%, NSBPSO is 84.888%, IMIDS is 86.888%, and XGB-RF is 87.877%. The TPR increased by the JS-DKN+ DSA is 3.30% better than the NSBPSO. Fig. 7 (c) illustrates the valuation of the JS-DKN+ DSA because of TNR. With 80% of the learning data, the JS-DKN+ DSA obtained a TNR of 87.765%, and TNR values of 77.988%, 79.877%, 81.876%, and 83.877% are attained by the traditional techniques such as DL-based ID, NSBPSO, IMIDS, and XGB-RF. The TNR of the JS-DKN+ DSA is improved by 4.43 % than the IMIDS.



(a) (b)

(c)

Fig. 6. *Appraisal of the JS-DKN+ DSA in view of learning data concerning a) accuracy, b) TPR, and c) TNR*

5.8. Performances Investigation

Fig. 8 exhibits the valuation of the JS-DKN+ DSA for the detection of intrusion concerning the learning data to several layers. Fig. 8 (a) epitomizes the appraisal of the JS-DKN+ DSA regarding the accuracy of the learning data. The JS-DKN+ DSA obtained an accuracy with the number of layers as 2,4,6, and 8 of 82.766%, 83.877%, 85.877%, and 86.755% for the learning data 60%. Fig. 8 (b) shows the valuation of the

JS-DKN+ DSA because of TPR. For 70% of the learning data, the JS-DKN+ DSA achieved by the TPR with 2, 4, 6, and 8 layers is 84.877%, 86.877%, 87.866%, and 90.877%. Fig. 8 (c) reveals the appraisal of the JS-DKN+ DSA based on the TNR. The TNR measured by JS-DKN+ DSA of 81.877%, 83.875%, 84.766%, and 87.765% for the number of layers as 2, 4, 6, and 8 regarding the learning data of 80%.

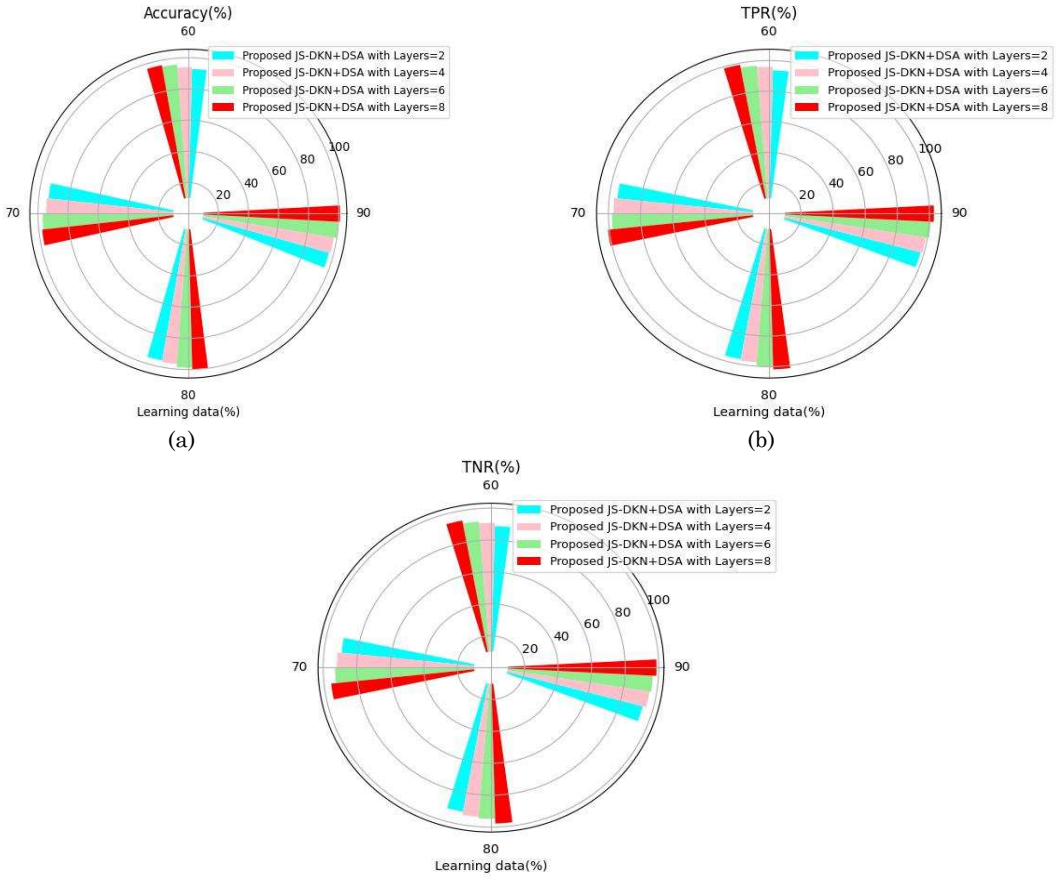


Fig. 7. *Performance's valuation of the JS-DKN+ DSA with respect to a) accuracy, b) TPR, and c) TNR*

5.9 Comparative Discussion

The JS-DKN+ DSA for feature fusion in IoT intrusion detection is correlated with several traditional methodologies and is illustrated in Table 1. The designed S-DKN+ DSA for detecting intrusion gained maximum TPR, accuracy, and TNR of 92.766%, 90.876%, and 88.654% with learning data of 90%. In addition, the accuracy achieved by the prevailing approach namely DL-based ID, NSBPSO, IMIDS, and XGB-RF is 82.877%, 85.877%, 86.877%, and 89.877%. Furthermore, the TPR recorded by the existing methods namely DL-based ID is 85.877%, NSBPSO is 87.877%, IMIDS is 89.877%, and XGB-RF is 90.877%. Moreover, the TNR gained by the DL-based ID is 79.877%, NSBPSO is 80.877%, IMIDS is 83.679%, and XGB-RF is 85.777%. The DKN provided a high accuracy and reduced the false positives. The DSA improves the scalability, and robustness, and reduces noise. Thus, the JS-DKN combines the most salient features resulting in reduced computation, and the detection of intrusion is carried out by the DSA, which enhances intrusion detection performance.

Table 1. *Comparative Discussion*

Metrics	***Techniques for intrusion detection***				
	DL-based ID	***NSBPSO***	***IMIDS***	***XGB-RF***	***Devised JS-DKN+ DSA***
Accuracy (%)	82.877	85.877	86.877	89.877	**90.876**
TPR (%)	85.877	87.877	89.877	90.877	**92.766**
TNR (%)	79.877	80.877	83.679	85.777	**88.654**

6. Conclusion

The network of physical objects like devices, vehicles, appliances, and other "things", which are combined with software, connectivity features, and sensors is known as the IoT. Intrusion detection in IoT has been used to control the system or network against policy violations and malicious activities. Moreover, IDS of high quality are more expensive to maintain and implement which is considered a barrier for organizations. Therefore, a new technique JS-DKN+ DSA is designed for feature fusion in IoT intrusion detection. Firstly, the simulation of IoT is considered, afterward, the input network traffic data are obtained from the dataset. Then, the z-score is applied for the data normalization. Subsequently, feature fusion is done using the JS-DKN. Lastly, IoT intrusion detection is carried out by using the DSA. Moreover, the established JS-DKN +DSA approach measured the highest TNR at 88.654%, accuracy of 90.876%, and a TPR of 92.766 %, which shows the superiority of the approach in detecting intrusions. In future work, a large number of datasets will be utilized to enlarge generalization and optimization techniques will be applied to improve the training process.

Compliance With Ethical Standards

Conflicts of interest: Authors declared that they have no conflict of interest.

Human participants: The conducted research follows ethical standards and the authors ensured that they have not conducted any studies with human participants or animals.

References

[1] A. Awajan, "A novel deep learning-based intrusion detection system for IOT networks", Computers, vol,12, no.2, pp.34, 2023.
[2] S. Baniasadi, O. Rostami, D. Martín, and M. Kaveh, "A novel deep supervised learning-based approach for intrusion detection in IoT systems", Sensors, vol.22, no. 12, pp.4459, 2022.
[3] K. H. Le, M. H. Nguyen, T. D. Tran, and N. D. Tran, "IMIDS: An intelligent intrusion detection system against cyber threats in IoT", Electronics, vol.11, no.4, pp.524, 2022.
[4] J. A. Faysal, S. T. Mostafa, J. S. Tamanna, K. M. Mumenin, M. M. Arifin, M. A. Awal, A. Shome, and S. S. Mostafa, "XGB-RF: A hybrid machine learning approach for IoT intrusion detection", In Telecom, Vol. 3, No. 1, pp. 52-69, January, MDPI, 2022.
[5] K. Albulayhi, Q. Abu Al-Haija, S. A. Alsuhibany, A. A. Jillepalli, M. Ashrafuzzaman, and F. T. Sheldon, "IoT intrusion detection using machine learning with a novel high performing feature selection method", Applied Sciences, vol.12, no.10, pp.5015, 2022.
[6] M. Zhong, Y. Zhou, and G. Chen, "Sequential model-based intrusion detection system for IoT servers using deep learning methods", Sensors, vol.21, no.4, pp.1113, 2021.
[7] M. Z. Al-Faiz, A. A. Ibrahim, and S. M. Hadi, "The effect of Z-Score standardization (normalization) on binary input due the speed of learning in back-propagation neural network", Iraqi Journal of Information and Communication Technology, vol.1, no.3, pp.42-48 b, 2018.
[8] L. Feng, and G. Yang, "Deep Kronecker Network", arXiv preprint arXiv:2210.13327, 2022.
[9] Y. Yu, J. Li, J. Li, Y. Xia, Z. Ding, and B. Samali, "Automated damage diagnosis of concrete jack arch beam using optimized deep stacked autoencoders and multi-sensor fusion", Developments in the Built Environment, vol.14, pp.100128, 2023.
[10] O. Elnakib, E. Shaaban, M. Mahmoud, and K. Emara, "EIDM: deep learning model for IoT intrusion detection systems", The Journal of Supercomputing, vol.79, no. 12, pp.13241-13261, 2023.
[11] S. Hajj, J. Azar, J. Bou Abdo, J. Demerjian, C. Guyeux, A. Makhoul, and D. Ginhac, "Cross-layer federated learning for lightweight IoT intrusion detection systems", Sensors, vol.23, no .16, pp.7038, 2023.
[12] A. R. Abdulla, and N.G.M. Jameel, "A review on IoT intrusion detection systems using supervised machine learning: Techniques, datasets, and algorithms", UHD Journal of Science and Technology, vol.7, no. 1, pp.53-65, 2023.
[13] N. Tekin, A. Acar, A. Aris, A. S. Uluagac, and V. C. Gungor, "Energy consumption of on-device machine learning models for IoT intrusion detection", Internet of Things, vol.21, pp.100670,2023.
[14] M. Catillo, A. Pecchia, and U. Villano, "CPS-GUARD: Intrusion detection for cyber-physical systems and IoT devices using outlier-aware deep autoencoders", Computers & Security, vol.129, pp.103210, 2023.
[15] F. Nielsen, "On a generalization of the Jensen–Shannon divergence and the Jensen–Shannon centroid", Entropy, vol.22, no.2, pp.221, 2020.
[16] The CIC IoT dataset 2023 is taken from, "https://www.unb.ca/cic/datasets/iotdataset-2023.html" accessed on 2024.