Received: June 4, 2023. Revised: July 9, 2023. 364

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Multi-Valued Neutrosophic Convolutional LSTM for Intrusion Detection

Veni Chinnasamy[1]* **Sridevi Rajasekaran[2]**

[1]*Department of Computer Science, PSG College of Arts and Science, Coimbatore, Tamilnadu, India*
[2]*Department of Computer Science with Data Analytics,*
PSG College of Arts and Science, Coimbatore, Tamilnadu, India
* Corresponding author's Email: venivijayan.c.phd@gmail.com

Abstract: Cyber security is an essential area of study because of the positive effects that networks have on modern society. As it becomes simpler for cybercriminals to launch novel assaults, the scale and complexity of today's networks continue to grow. Therefore, it is crucial to create an efficient instruction detection system (IDS) capable of constantly monitoring network traffic for suspicious activity and issuing alerts as necessary. Several researchers have employed machine learning (ML) and deep learning (DL) techniques to create an effective IDS. Amongst, an effective IDS detection model convolutional neural network (CNN)- long short term memory (LSTM extracts temporal and geographical aspects of network traffic to lower the false alarm rate (FAR) and raise the detection rate (DR). However, the epistemic uncertainty in intrusion data affect the efficiency of CNN-LSTM model. This paper handle uncertainty information by proposing a multivalued neutrosophic convolutional LSTM (MVN-ConvLSTM) which extracts different deep features. MVN-ConvLSTM uses a neutrosophic set (NS) which considers truth (T), indeterminacy (I), and falsity (F) memberships of every features. First, the intrusion features are mapped to NS domain as six sets, positive degree of truth-membership (T_A), positive degree of falsity membership (F_A), Positive degree of indeterminate towards falsity and indeterminate towards truth membership (I_A), negative degree of truth-membership (T_B), negative degree of falsity membership (I_B), and negative degree of indeterminate towards falsity and indeterminate towards truth membership (I_B). Next, an MVN-ConvLSTM is generated with four parallel paths, two of which take input from T and two from I, and a suitable combination of these paths is used to produce the output. The weights of the neural network are updated in four different directions at once using a back propagation approach. The efficiency of MVN-ConvLSTM to handle uncertainty in intrusion dataset is proved by comparing conventional CNN models which provides lower FAR and high DR. Finally, the test results show that the MVN-ConvLSTM achieves accuracy value of 94.63%, 92.84% and 91.84% on three different datasets CIC-IDS2018, WSN-DS and UNSW-NB15.

Keywords: Network security, Machine learning, Deep learning, Neutrosophic set, Long short term memory.

1. Introduction

The internet of things (IoT), big data, and cloud computing, as well as growing reliance on interconnected network need relevant network security to handle vulnerabilities and threats [1]. Traditional security methods such as firewalls and encryption approaches face challenges where the attackers continue to build increasingly complex attacks [2]. In addition, researchers in the field of cyber-security recognized the significance of constructing effective network IDS to secure the networks from malicious activities and unauthorised access [3].

IDS system mostly developed with the goal of recognise identified, unidentified attacks and threats with high accuracy and a low FAR [4]. Misuse detection and anomaly detection are two methods of IDS. The misuse detection or signature-based detection relies on the existence of known attacks and threats relies on unknown attacks [5]. As anomaly detection tends to detect both known and unknown attacks. The IDS methods more susceptible for detecting unknown new attacks while networks size and offered services are grown [6, 7]. To guarantee

the security of such networks, an IDS must be capable of detecting and mitigating both known and unknown threats.

IDS utilizes the benefits of AI to understand form the IDS datasets without any human interactions. Generally, ML and DL methods of AI are used to categorize network traffics. When compared to ML, DL has high model's recognition accuracy [8]. DL models such as CNN, auto-encoders (AE), and recurrent neural networks (RNN), deep belief networks (DBN) are used for [9]. Hybrid IDS model like CNN and long-short term memory (LSTM) are used in [10]. In order to reduce the FAR and increase the DR, both CNN and LSTM capture the spatial and temporal aspects of network traffic. This approach is further improved by the use of batch normalisation and dropout layers. The performance of DL models used for intrusion detection is significantly impacted by the availability of epistemic uncertainties in IDS datasets.

Therefore, in this paper, MVN-ConvLSTM is proposed to handle the uncertainty information's from IDS datasets. The NS domain which is an extension of the fuzzy set is used in this model to consider the T, I, and F memberships in an effort to find a solution. The proposed technique first maps the incursion features from the feature domain to three sets in the NS domain (T, I, and F). Again, map the features as the T_A, F_A, I_A, T_B, F_B and I_B using the obtained positive and negative degree of membership values from NS domain before moving to the DL procedure.

So, four parallel routes using MVN-ConvLSTM is developed using two T input and two I input, and then select the optimal combination of routes to obtain the desired output. Using the back propagation approach, four paths is simultaneously trained while adjusting the weights of the neural network. Additionally, during the initial epoch of training, the weights of four routes are changed simultaneously by multiplying the path outputs, leading to gradient switching. The paths interact and aid one another along this axis, causing significant shifts in overall weight. The constructed ConvLSTM structure can learn effectively even the input sample have significant variations. The weight updating among four path make each path to learn more effectively which can provide better results for classes having small instances. The efficiency of MVN-ConvLSTM in dealing with uncertainty is demonstrated for different datasets CIC-IDS2018, WSN-DS, and UMSW-NB15.

The remaining article is prepared as follows: Section 2 presents the studies associated with the prediction of intrusion detection using DL algorithms. Section 3 discusses the proposed algorithm and Section 4 shows its performance compared to the existing algorithms. Section 5 concludes this study and suggests future enhancements.

2. Literature survey

As an IDS method, Conv-LSTM was proposed [11] with the One-Hot coding as a pre-processing with the aid of normalization. Then, the LSTM network topology was modified to include a convolution operation, which enhances the reliability of network intrusion detection. However, this method has low detection rate for rare type attack classes which limits the overall accuracy results.

The resilient IDS based on the DL-generative adversarial network (GAN) model for effective IDS and improved edge network security [12]. For feature learning, this technique employs a GAN model to recover low-dimensional features from original network flows. The feature selection mechanism was first employed to handle collaborative edge network data. Then, for IDS aimed against a single assault, a DL architecture based on GAN was devised. Lastly, a multi-capability IDS model was built to identify new forms of assaults. However, this method has low detection rate for low frequent attack types.

A data fusion technique was designed to fuse data from several inputs in order to acquire more useful and precise data [13]. The relational algebra left join approach was used for data fusion. The K-neighbor nearest (KNN) method was then employed in conjunction with ensemble learning to improve generalizability by combining inadequate learners and transforming them into robust learners. This model use data fusion models to identify unexpected forms of assaults with lower FAR findings. However, higher number of cross validation only yield better precision results.

An integrative deep IDS model using stacked denoising auto-encoder (AE) - extreme learning machine (SDAE-ELM) [14] to predict intrusion behaviour. To speed up intrusion detection, this model makes use of both SDAE and ELM; the former filters out data and network noise, while the latter speeds up training. The model's categorization accuracy was then enhanced by using DBN models. However, this model has high accuracy results for complex structure and utilizing large number of parameters.

In order to differentiate between malicious and safe sessions on a network, one-class classifier with the support of GAN [15] model analyses packet data. The misclassification of data by the use of DNN-LSTM trained model which is trained from the

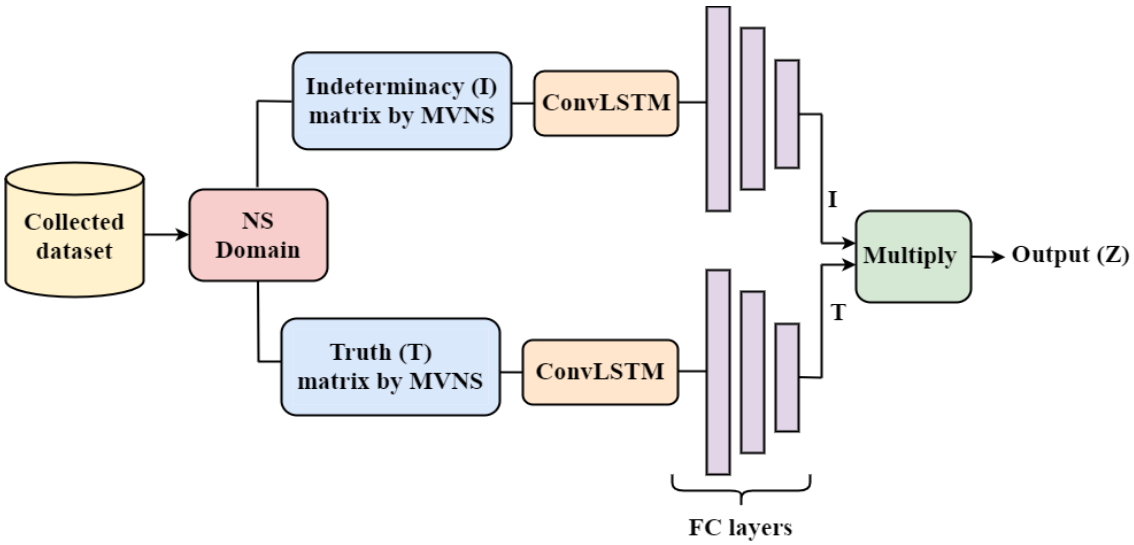

Figure. 1 Schematic representation of MVN-ConvLSTM mode

original data is corrected by instances created by GAN. The GAN model might determine the current packet has a good chance of being correctly classified by the DNN-LSTM. Although the proposed approach has a fast detection rate, accuracy has to be enhanced further.

A DNN model was suggested for RT-IDS [16]. The needs of a real-time IDS can be met by this paradigm. Deployed between the gateway router and the LAN, a real-time feature extractor snoops on live network traffic in order to extract useful information about the features of the gathered dataset. In order to identify malicious actions in the network, the ML pipeline was hosted on a server alongside the trained DNN model and is accessible through API. However, the accuracy of the model relied heavily on the training, and system uncertainty was significant.

A new anomaly-based IDS model was [17] combining PSO with the grey wolf optimisation (GWO) model. Initially, the acquired dataset was mined for highly correlated characteristics using the GA-based technique to enhance this model's detection capability. Then, a BPNN was generated with a PSO-GWO algorithm hybrid. Finally, this combined approach was applied to the original dataset to resolve binary and multi-class classification issues. However, this paradigm was vulnerable to undiscovered or very modest problems.

In order to create an IDS, [18] DL and a variant of the reptile search algorithm (RSA) was proposed. In this model, feature extraction and selection were accomplished using DL and MH optimisation techniques. After then, CNNs were utilised as the feature extractor to develop more accurate and informative representations of the input data in a reduced dimensional space. To improve the IDS system's efficiency, a novel RSA-based feature selection process was developed to narrow down the retrieved features using the CNN model to only the most crucial ones. But, the proper training model should be maintained for transfer learning to obtain high accuracy results.

3. Proposed methodology

In this section, the MVN-ConvLSTM model is briefly illustrated. The Fig. 1 depicts the schematic representation of the proposed work. Table 1 defines the notation list.

3.1 Multi-valued neutrosophic sets (MVNSs)

Researchers have showed a strong interest in hesitant fuzzy sets (HFSs) [19] and NSs, which have been frequently used in multi-attribute decision-making situations in recent years. In this architecture, MVNSs are utilized to handle the issue of uncertainty information in intrusion data and aid to boost the DR. Assume that X is a space points (intrusion feature), and that a generic element in X is designated by.

A MVNSs M in X is defined by three functions like $\check{T}_M(a)$, $\check{I}_M(a)$ and $\check{F}_M(a)$ in the range of [0, 1], as shown in Eq. (1).

$$M = \{< a, \check{T}_M(a), \check{I}_M(a) \text{and } \check{F}_M(a) > \mid a \in X\} \quad (1)$$

Where, $\check{T}_M(a)$, $\check{I}_M(a)$ and $\check{F}_M(a)$ are NS sets with values in [0,1], $T-$, $I-$ and $F-$ membership degree respectively with satisfying some conditions which are listed below.

(i) $0 \leq \gamma, \eta, \xi \leq 1,\ \ 0 \leq \gamma^+ + \eta^+ + \xi^+ \leq 3$

Table 1. Notation list

X	Intrusion Features
$\check{T}_M$	Truth Membership
$\check{I}_M$	Indeterminacy Membership
$\check{F}_M$	Falsity Membership
W_{I_A} and W_{I_B}	Assigned weights of $I-path$
W_{T_A} and W_{T_B}	Assigned weights of $T-path$
$s(M)$	Scoring Function
$a(M)$	Accuracy Function
$s(M)$	Scoring Function
$c(M)$	Certainty Function
$X_1, X_2, X, \ldots\ldots, X_n$	Input states in ConvLSTM
$C_1, C_2, C_3, \ldots\ldots, C_n$	Cell states in ConvLSTM
$H_1, H_2, H_3, \ldots\ldots, H_n$	Hidden states in ConvLSTM
I_t	Input gate
O_t	Output gate
C_t	Cell gate
'$*$'	Convolution
'$\times$'	Hadamard Product
η	Learning rate
I_A & I_B	$I-$ path
T_A & T_B	$T-$ path

Where, $\gamma \in \check{T}_M(a)$, $\eta \in \check{I}_M(a), \xi \in \check{F}_M(a)$, $\gamma^+ = sup\check{T}_M(a)$, $\eta^+ = sup\check{I}_M(a)$ and $\xi^+ = sup\check{F}_M(a)$

If X has only one NS, then M is simply as NS, otherwise it is multi-valued neutroscophic numbers (MVNNs) $M = \langle \check{T}_M(a), \check{I}_M(a), \check{F}_M(a) \rangle$. For convenience, an MVNN can be termed as $M = \langle \check{T}_M, \check{I}_M, \check{F}_M \rangle$.

Obviously, MVNSs are generally considered as an extension of NSs. An MVNN can be written as $M = \langle \check{T}_M, \check{I}_M, \check{F}_M \rangle$ for convenience.

If each of $M = \langle \check{T}_M, \check{I}_M, \check{F}_M \rangle$ values are found by the condition i.e., $\gamma, \eta\ \xi$ and $0 \le \gamma + \eta + \xi \le 3$, correspondingly, then MVNSs are minimised to Singled-NSs (SVNSs) [20].

ii) If $\check{T}_M(a), \check{I}_M(a), \check{F}_M(a)$ are all interval values then MVNSs are lowered to internal valued NSs [21]. If $\check{T}_M(a) = \phi$ for any a, then MVNSs are switched to Dual HFSs (DHFSs) [22].

(iii) If $\check{T}_M(a) = \check{F}_M(a) = \phi$ for any a. Thus, MVNSs are merely expansions of HFSs [19]. The complement of MVNSs M^c is mentioned in Eq. (2),

$$M^c = \langle \cup_{\gamma_M \in \check{T}_M}\{1-\gamma_M\}, \cup_{\eta_M \in \check{I}_M}\{1-\eta_M\}, \cup_{\xi_M \in \check{F}_M}\{1-\xi_M\} \rangle \quad (2)$$

Then, consider $M = \langle \check{T}_M, \check{I}_M, \check{F}_M \rangle$ and $N = \langle \check{T}_N, \check{I}_N, \check{F}_N \rangle$ are two MVNNs, $M \prec N$ if and only if $\forall\ \check{T}_M^x \in T_M$, $\check{T}_N^y \in T_N$, $\check{I}_M^x \in I_M$, $\check{I}_N^y \in I_N$, $\check{F}_M^x \in F_M$, $\check{F}_N^y \in F_N$, and $\check{T}_M^x < \check{T}_N^y, \check{I}_M^x < \check{I}_N^y, \check{F}_M^x < \check{F}_N^y$.

The function of s(M), a(M), and c(M) of an MVNN are represented in Eq. (3), Eq. (4), and Eq. (5) represent the score values of NS and accuracy values of NS and certainty values of NS

$$s(M) = \frac{1}{l_{\check{T}_M} \cdot l_{\check{I}_M} \cdot l_{\check{F}_M}} \times \sum_{\gamma_i \in \check{T}_M, \eta_i \in \check{I}_M, \xi_i \in \check{F}_M} (\gamma_i + 1 - \eta_j + 1 - \xi_k)/3 \quad (3)$$

$$a(M) = \frac{1}{l_{\check{T}_M} \cdot l_{\check{F}_M}} \times \sum_{\gamma_i \in \check{T}_M, \xi_i \in \check{F}_M} (\gamma_i - \xi_k) \quad (4)$$

$$c(M) = \frac{1}{l_{\check{T}_M}} \times \sum_{\gamma_i \in \check{T}_M} (\gamma_i) \quad (5)$$

Where, $\gamma_i \in \check{T}_M$, $\eta_i \in \check{I}_M$, $\xi_i \in \check{F}_M$, denotes the elements number and $l_{\check{T}_M}$, $l_{\check{I}_M}$, and $l_{\check{F}_M}$ are the like-hood preferences in $\check{T}_M$, $\check{I}_M$, $\check{F}_M$ respectively.

When evaluating MVNNs, the $s(M)$ index is crucial. If M is an MVNN, then T – membership is a larger subset of $(\check{T}_M)$. And the I membership $(\check{I}_M)$ is less, the MVNN is greater. Similarly, the $F-$ membership $(\check{F}_M)$ is smaller, the MVNN is greater. If the difference between T and F bigger and then the statement is more affirmative, the $a(M)$ is satisfied. That is, the larger the values of $\check{T}_M$, $\check{I}_M$ and $\check{F}_M$, the more the accuracy of the MVNN. As to the certainty function, the value of T- membership is bigger, it means more certainty of the MVNNs.

3.2 ConvLSTM structure

The LSTM's cell state c_t is the most crucial component since it is where data is stored. Input value is saved if input gate i_t is activated, while prior state c_{t-1} is forgotten if forget gate f_t is activated. In addition, the conversion from the current cell state c_t to the ultimate hidden state h_t is determined by the output cell o_t. The simplest LSTM models function in this way. In contrast, ConvLSTM layer has 3D tensors for all inputs $X_1, X_2, X, \ldots\ldots, X_n$, cell states $C_1, C_2, C_3, \ldots\ldots, C_n$, hidden states $H_1, H_2, H_3, \ldots\ldots, H_n$, and gates I_t, C_t, O_t. The inputs and gates of the ConvLSTM layer are first modelled as vectors in a grid-like structure in space. Fig. 2 depicts the cell layout of the ConvLSTM.

The ConvLSTM layer takes the inputs and the current state of the local entities in a given cell and outputs a prediction for the cell's future state. Examining the subsequent Eqs. (6-10) will help to clarify the situation.

$$F_t = \sigma(W_{XF} * X_t + W_{HF} * H_{t-1} + W_{CF} * C_{t-1} + b_f) \quad (6)$$

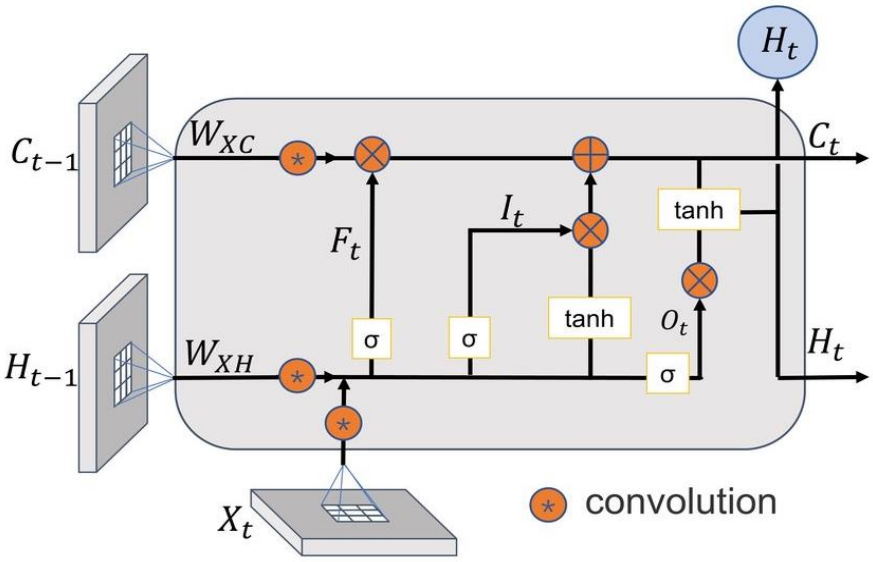

Figure. 2 structure of ConvLSTM cell

$$I_t = \sigma(W_{XI} * X_t + W_{HI} * H_{t-1} + W_{CI} * C_{t-1} + b_i) \quad (7)$$

$$O_t = \sigma(W_{XO} * X_t + W_{HO} * H_{t-1} + W_{CO} * C_{t-1} + b_o) \quad (8)$$

$$C_t = F_t \times C_{t-1} + I_t \times \tanh(W_{XC} * X_t + W_{HC} * H_{t-1} + b_c) \quad (9)$$

$$h_t = O_t \times \tanh(C_t) \quad (10)$$

The weight matrices that are being investigated are $W_{XF}, W_{XI}, W_{XO}, W_{XC}, W_{CF}, W_{CI}, W_{CO}$ and W_{HC}. The four biases are b_f, b_i, b_o and b_c. Each up-versed will include a full recalculation of all weight matrices and bias vectors.

3.3 MVN-ConvLSTM model

In this setup, the presentation of I and T sets is considered for ConvLSTM structures resulting in a two-parallel-path network is defined as MVN-ConvLSTM. Both the positive and negative degrees of I-path i.e., I_A and I_B, and T-path i.e., T_A and T_B, are trained in the first and second routes, correspondingly. All of these paths contribute to the network's final projected label. Unlike traditional two-path architectures, which train each path independently before combining their accumulated weights at the end, this one trains both paths concurrently. Additionally, the weights of the two pathways are adjusted simultaneously during the first epoch of training by multiplying the outputs of the paths, leading in gradient switching. In this case, trustworthy weight updates are the result of inter-linked routes that mutually support and reinforce one another.

Fig. 3 depicts four weights, two in the $I-path$ as W_{I_A} and W_{I_B} and two in the $T-path$ as W_{T_A} and W_{T_B}. The Fig. 3 module clearly shows that the responses of the neurons in the previous layer are multiplied by these weights in order to produce I_A and I_B with T_A and T_B Labels are integrated (multiplied) to get G_A and G_B as the resultant outcome.

This framework demonstrates how to update the weights on the $T-$path (W_{T_A} and W_{T_B}) and the same idea is applied to the $I-$ path (W_{I_A} and W_{I_B}). The generic update rule for a $T-$path network's weights is provided by Eq. (11) and Eq. (12), and it applies to both the weight domain W_{T_A} and W_{T_B} is constructed as follows

$$W_{T_A} = W_{T_A} + \Delta W_{T_A} \quad (11)$$

$$W_{T_B} = W_{T_B} + \Delta W_{T_B} \quad (12)$$

Where, ΔW_{T_A} and ΔW_{T_B} is calculated based on the neural network updated rule in Eq. (13) & Eq. (14)

$$\Delta W_{T_A} = -\eta . \frac{\partial N_A}{\partial W_{T_A}} \quad (13)$$

$$\Delta W_{T_B} = -\eta . \frac{\partial N_B}{\partial W_{T_B}} \quad (14)$$

Applying chain rule to calculate $\frac{\partial N}{\partial W_{T_A}}$ & $\frac{\partial N}{\partial W_{T_B}}$ leads to

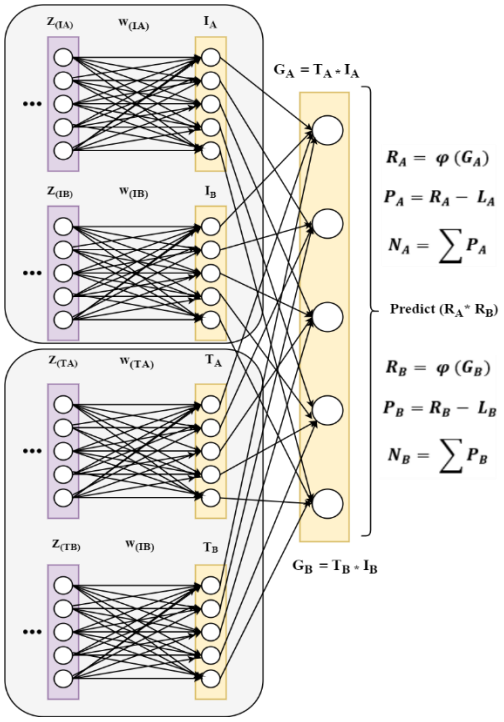

Figure. 3 Network weights for MVN-ConvLSTM

$$\frac{\partial N_A}{\partial W_{T_A}} = \frac{\partial N_A}{\partial G_A}.\frac{\partial G_A}{\partial T_A}.\frac{\partial T_A}{\partial W_{T_A}} \quad (15)$$

$$\frac{\partial N_B}{\partial W_{T_B}} = \frac{\partial N_B}{\partial G_B}.\frac{\partial G_B}{\partial T_B}.\frac{\partial T_B}{\partial W_{T_B}} \quad (16)$$

Based on network structure in Fig. 3, G_A and G_B with T_A and T_B are calculated by Eq. (17) & Eq. (18)

$$G_A = T_A . I_A \quad \text{and } T_A = \sum w_{T_A} . Z_{T_A} \quad (17)$$

$$G_B = T_B . I_B \text{ and } T_B = \sum w_{T_B} . Z_{T_B} \quad (18)$$

Therefore,

$$\frac{\partial G_A}{\partial T_A} = I_A \text{ and } \frac{\partial T_A}{\partial w_{T_A}} = Z_{T_A} \quad (19)$$

$$\frac{\partial G_B}{\partial T_B} = I_B \text{ and } \frac{\partial T_B}{\partial w_{T_B}} = Z_{T_B} \quad (20)$$

By substituting the Eq. (19) in Eq. (15), the Eq. (21) is constructed as follows

$$\frac{\partial N_A}{\partial W_{T_A}} = \frac{\partial N_A}{\partial G_A} . I_A . Z_{T_A} \quad (21)$$

Similarly, by substituting the Eq. (20) in (16), the Eq. (22) is constructed as follows

$$\frac{\partial N_B}{\partial W_{T_B}} = \frac{\partial N_B}{\partial G_B} . I_B . Z_{T_B} \quad (22)$$

To compute the $\frac{\partial N_A}{\partial G_A}$ & $\frac{\partial N_B}{\partial G_B}$ chain rule is utilized again which tends to;

$$\frac{\partial N_A}{\partial G_A} = \frac{\partial N_A}{\partial P_A} . \frac{\partial P_A}{\partial R_A} . \frac{\partial R_A}{\partial G_A} \quad (23)$$

$$\frac{\partial N_B}{\partial G_B} = \frac{\partial N_B}{\partial P_B} . \frac{\partial P_B}{\partial R_B} . \frac{\partial R_B}{\partial G_B} \quad (24)$$

R_A and R_B in Eq. (23) and (24), respectively, indicates the final network output, whereas P_A and P_B indicate the difference between the data label L_A and L_B and the forecasted label R_A and R_B, respectively. Total error squared L_A and L_B and Eqs. (25) and (26) define these parameters

$$P_A = R_A - L_A(label), N_A(n_A) = \frac{1}{2} \sum {P_A}^2(n_A) \quad and\ R_A = \varphi(G_A) \quad (25)$$

$$P_B = R_B - L_B(label), N_B(n_B) = \frac{1}{2} \sum {P_B}^2(n_B) \quad and\ R_B = \varphi(G_B) \quad (26)$$

In application of derivation rules, Eq. (27) & Eq. (28) is depicted as follows,

$$\frac{\partial N_A}{\partial P_A} = P_A \text{ , } \frac{\partial P_A}{\partial R_A} = 1 \text{ and } \frac{\partial R_A}{\partial G_A} = \varphi'(G_A) \quad (27)$$

$$\frac{\partial N_B}{\partial P_B} = P_A \, , \frac{\partial P_B}{\partial R_B} = 1 \text{ and } \frac{\partial R_B}{\partial G_B} = \varphi'(G_B) \quad (28)$$

Therefore, $\frac{\partial N_A}{\partial G_A}$ and $\frac{\partial N_B}{\partial G_B}$ is computed in Eq. (29) & Eq. (30)

$$\frac{\partial N_A}{\partial G_A} = P_A . \varphi'(G_A) \quad (29)$$

$$\frac{\partial N_B}{\partial G_B} =. P_B . \varphi'(G_B) \quad (30)$$

The update rule is executed in Eq. (31) and Eq. (32) by substituting Eq. (29) with Eq. (21).

$$\frac{\partial N_A}{\partial W_{T_A}} = P_A . \varphi'(G_A) . I_A . Z_A \quad (31)$$

$$W_{T_A} = -\eta . P_A . \varphi'(G_A) . P_A . Z_A \quad (32)$$

The update rule is implemented in Eq. (33) & (34) by replacing Eq. (30) with Eq. (22).

$$\frac{\partial N_B}{\partial W_{T_B}} = P_B . \varphi'(G_A) . I_B . Z_B \quad (33)$$

$$W_{T_B} = -\eta . P_B . \varphi'(G_B) . P_B . Z_B \quad (34)$$

According to the above equations, weight updates for neurons on the T-path (T_A & T_B) are impacted by I_A & I_B subset in the MVNN domain, and vice versa. As a conclusion, it is shown that MVN-ConvLSTM produces higher quality output than regular CNN.

The parameters I_A & I_B are the primary differentiators between the weight update rules of NCNN and regular CNN. The following example shows how to extend Eq. (31) to Eq. (32) and Eq. (33) to Eq. (34) to calculate the impact of I_A & I_B on the derivation of Δw_{T_A} and Δw_{T_B} , correspondingly. This means that L_A and L_B can be assigned either the value 0 or the 1. Consequently, the values in Eq. (37) and Eq. (38) may be treated in Eq. (35) and Eq. (36) independently as,

$$\begin{aligned} w_{I_A} &= [-\eta . (R_A - L_A) . \ \varphi'(G_A) . I_A . y] \\ &\approx [-\eta . (-L_A) . \ \varphi'(G_A) . I_A . y] \approx [(G_A - L_A) . I_A] = (G_A - L_A) . I_A] \\ &= |T_A . I_A - L_A| . I_A = |T_A . (I_A)^2 - L_A| . I_A \end{aligned} \quad (35)$$

$$\begin{aligned} \Delta w_{I_B} &= [-\eta . (R_B - L_B) . \ \varphi'(G_B) . I_B . y] \\ &\approx [-\eta . (-L_B) . \ \varphi'(G_B) . I_B . y] \\ &\approx [(G_B - L_B) . I_B] = (G_B - L_B) . I_B] \\ &= |T_B . I_B - L_B| . I_B \\ &= |T_B . (I_B)^2 - L_B| . I_B \end{aligned} \quad (36)$$

$$\begin{cases} |I_A - T_A . (I_A)^2| & if \ L_A = 1 \\ |T_A . (I_A)^2| & if \ L_A = 0 \end{cases} \quad (37)$$

$$\begin{cases} |I_B - T_B . (I_B)^2| & if \ L_{\boldsymbol{B}} = 1 \\ |T_B . (I_B)^2| & if \ L_B = 0 \end{cases} \quad (38)$$

T_A and I_A with T_B and I_B are the output of the softmax layer, and hence these parameters are valued as 0 and 1, as seen in the comparison between the MVN-ConvLSTM and the traditional CNN in Eq. (39) and Eq.(40).

$$\begin{cases} |1 - T_A|, & if \ L_A = 1 \\ |T_A|, & if \ L_A = 0 \end{cases} \quad (39)$$

$$\begin{cases} |1 - T_B|, & if \ T_B = 1 \\ |T_B|, & if \ T_B = 0 \end{cases} \quad (40)$$

The main challenge in the traditional CNN model, it is very much sensitive towards the uncertainty data. In other words, network weights are incrementally changed when clean data are provided as input and their label is accurately predicted. The network tries to significantly update weights when fed the same data (with the same label) with uncertainty. In tradition CNN methods, the weight update for clean and uncertainty data with the same label is significantly different which leads to misinterpret the network with uncertainty data. In this framework, it is demonstrated that MVN-ConvLSTM handles the uncertainty data efficiently and does not update weights for clean and uncertainty data with the same label in a different way. While comparing the amount of weight updated for clean and uncertainty data in MVN-ConvLSTM and CNN model, if true label 1 is taken into account for noisy and clean data, weight updates for clean $(1 - T_A(c))$ and $(1 - T_B(c))$ and uncertainty data in CNN are $(1 - T_A(c))$ and $(1 - T_B(c))$, respectively.

The fundamental problem with the standard CNN model is that it is extremely susceptible to the uncertainty data. In other words, when good data are used as input and their label is correctly expected, the network weights are modified slightly. When the network is given identical data (labelled the same) with uncertainty, it attempts to dramatically adjust the weights. In conventional CNN approaches, misinterpretation of the network takes place when dealing with uncertainty data since the weight update for clean and uncertainty data with the same label is drastically different.

Within this structure, it is shown that MVN-ConvLSTM effectively manages uncertainty data and does not perform separate weight updates on clean and uncertainty data that share the same label. If true label 1 is considered for both chaotic and clean data, then the CNN model's weight adjustments for clean data are $(1 - T_A(c))$ and $(1 - T_B(c))$. Thus, Eq. (41) and Eq. (42) show how weight adjustments in CNN change when dealing with clean and uncertainty data.

$$T_B(c) - T_B(u) \tag{41}$$

$$T_B(c) - T_B(u) \tag{42}$$

In addition, the weight update for apparent and uncertainty data in MVN-ConvLSTM is calculated as $(I_A(c) - T_A(c) \,.\, (I_A\,(c)^2))$ and $(I_B(c) - T_B(c) \,.\, I_B\,(c)^2))$ with $(I_A(u) - T_A(u) \,.\, (I_A\,(u)^2))$ and $(I_B(u) - T_B(u) \,.\, (I_B\,(u)^2))$ respectively. The differences among the weight modification for clean and noisy data in MVN-ConvLSTM is given in Eq. (43) & Eq. (44).

This fact that MVN-ConvLSTM does not vary in weight update for clean and chaotic data compared to conventional CNN may be deduced from Eq. (42) & Eq. (43), which shows that Eq. (43) & Eq. (44)

$$\begin{aligned}&(I_A(u) - (T_A\,(u).\ (I_A\,(u))^2) - ((I_A(c) - \\ &\left(T_A\,(c).\left(\,I_A\,(c)\right)^2\right) = \ \left(I_A(c) - I_A\,(u)\right) + \\ &(\left(T_A\,(c).\left(\,I_A\,(c)\right)^2\right) - \left(T_A\,(u).\left(\,I_A\,(u)\right)^2\right)\,)\end{aligned} \tag{43}$$

$$\begin{aligned}&(I_B(u) - (T_B\,(u).\ (I_B\,(u))^2) - ((I_B(c) - \\ &\left(T_B\,(c).\left(\,I_B\,(c)\right)^2\right) = \ \left(I_B(c) - I_B(u)\right) + \\ &(\left(T_B\,(c).\left(\,I_B(c)\right)^2\right) - \left(T_B(u).\left(\,I_B\,(u)\right)^2\right)\end{aligned} \tag{44}$$

is less than Eq. (41) & Eq. (42). In other words, MVN-ConvLSTM is better able to deal with uncertainty in a reliable manner.

Moreover, it is obvious that $I_A(c)$and $I_B(c)$ with $T_B(c)$ and $T_B(c)$ are greater than $I_A(u)$ and $I_B(u)$ with $T_B(u)$ and $T_B(u)$ for a data point with label 1. Because, $I_A(c) - T_A(c)$; $I_B(c) - T_B(c)$; $I_A(c) - T_A(c)$; $I_B(c) - T_B(c)$ insists to a negative value, demonstrating that Eq. (43) & Eq. (44) is lesser that Eq. (45) & Eq. (46) respectively, which is depicted as

$$\begin{aligned}&((T_A\,(c).\,(I_A\,(c))^2) - (T_A\,(u).\,(I_A\,(u))^2) < \\ &((T_A\,(c) - T_A\,(u))) + (((I_A\,(c) - (I_A\,(u))))\end{aligned} \tag{45}$$

$$\begin{aligned}&((T_B\,(c).\,(I_B\,(c))^2) - (T_B\,(u).\,(I_B\,(u))^2) < \\ &((T_B\,(c) - T_B\,(u))) + ((\left(I_B\,(c) - \left(I_B\,(u)\right)\right)\end{aligned} \tag{46}$$

When trained independently, the I-path and T-path networks nearly always make the similar prediction for the target label. As a result, elements in the pairs $T_A\,(c)$ and $T_B\,(c)$ with $I_A\,(c)$ and $I_B\,(c)$, as well as $T_A\,(u)$ and $T_B\,(u)$ with $I_A\,(u)$ and $I_B\,(u)$ have close values. The I-path learns labels from uncertainty data points with a strong gradient, but T-path learns labels from all data patterns. As a result, in MVN-ConvLSTM, both uncertainty data points (high I) and clean data points are used to predict the label. From the description, it is analyzed that the proved MVN-ConvLSTM model effectively handles the uncertainty issues and provides a good performance for intrusion prediction in network system.

4. Result and discussion

Three different datasets used in this paper are listed below.

CIC-IDS2018 [23]: The communications security establishment (CSE) and the canadian institute for cybersecurity (CIC) worked together on a project to systematically develop a cybersecurity dataset using the concept of profiles, and this dataset is the product of their efforts. Seven distinct types of assaults like Brute force, Heartbleed, Botnet, denial of service (DoS), distributed denial of service, Web attacks, and internal infiltration are represented in the dataset. The 50-machine assaulting structure confronts a target with 420 workstations and 30 servers across 5 divisions. In addition to 80 network traffic characteristics derived from collected traffic using CICFlowMeter-V3, this collection also contains the attacker's side network traffic and log records from each computer.

WSN-DS [24]: WSN-DS was created in 2016 to track the number of nodes in wireless networks using detectors in order to detect regular and destructive traffic. The LEACH routing procedure, indicated by 23 features, is employed to retrieve 170 records from this dataset. Four different forms of Denial of Service (DoS) attacks exist; they are known as "standard records," "fooding attacks," "Grayhole attacks," and "Time Division Multiple Access" (TDMA).

UNSW-NB15 [25]: Records of both innocuous traffic and nine types of assaults (including Fuzzers, analysis, Backdoor, DoS, 160 Exploits, etc.) are

Table 1. Dataset details

Datasets	Samples	Features
CIC-IDS2018	261147	30
WSN-DS	68602	18
UNSW-NB15	175341	45

Table 2. Confusion matrix of existing and proposed models for validation set

Dataset\ methods		Conv-LSTM [11]	SDAE-ELM [14]	GAN -DNN-LSTM [15]	CNN-RSA [18]	CNN-LSTM [10]	MVN-ConvLSTM [Proposed]
C I C-IDS2018	TP	33045	33065	33606	33626	31760	31780
	TN	14254	14274	15008	15028	17625	17645
	FP	2769	2749	2083	2063	1422	1403
	FN	2162	2142	1533	1513	1423	1402
WSN-DS	TP	3942	3977	4007	4027	4078	4098
	TN	8480	8467	8553	8574	8625	8645
	FP	607	594	527	520	469	449
	FN	691	683	620	600	549	529
UNSW-NB15	TP	10788	10838	10863	10893	10908	10933
	TN	21079	21129	21204	24234	21249	21274
	FP	1651	1601	1551	1491	1476	1451
	FN	1611	1561	1511	1451	1436	1411

included in this dataset. In 2015, it was established by the Australian centre for cyber security (161). Information was gathered from 162 three authentic websites: BID (Symantec corporation), CVE (common vulnerabilities and exposures), and MSD (Microsoft corporation) (Microsoft Security bulletin).

Table 1 represent the dataset used in this paper. The quantity of samples and features for three datasets are listed. The acquired data is divided into a training set (80%) and a validating set (20%).

The proposed MV-ConvLSTM and other algorithms such as Conv-LSTM [11], SDAE-ELM [14], GAN-DNN-LSTM [15], CNN RSA [10] and CNN-LSTM [18] are developed by Python 3.7.8 for same three intrusion datasets. The evolution metrics like accuracy, precision and recall are evaluated. Table 2 displays the confusion matrices for the proposed and current frameworks.

4.1 Accuracy

The accuracy is measured as the number of accurate intrusion data classifications into either attack type or normal records divided by the total number data samples provided to the model which is formulated in Eq. (47),

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (47)$$

TP indicates that an intrusion instance is correctly classified as an intrusion, FP indicates that the intrusion instance is incorrectly classified as a normal instance, TN indicates that a normal instance is correctly classified as a normal instance, and FN indicates that the normal instance is incorrectly classified as an intrusion.

Fig. 4 provides a scatter plot showing the accuracy (%) of the presented and existing IDS for collected datasets. It indicates that the accuracy of MVN- ConvLSTM is 4.49%, 4.40%, 1.67%, 1.59% and 0.08% higher than Conv-LSTM, SDAE-ELM, DNN-LSTM, CNN-RSA and CNN-LSTM models on CIC-IDS2018. Similarly, the precision of MVN-ConvLSTM is 2.57%, 2.40%, 1.35%, 1.12% and 0.31% higher than Conv-LSTM, SDAE-ELM, DNN-LSTM, CNN-RSA and CNN-LSTM models on WSN-DS. The precision of MVN-ConvLSTM is 1.25%, 0.92%, 0.61%, 0.25% and 0.15% higher than Conv-LSTM, SDAE-ELM, DNN-LSTM, CNN-RSA and CNN-LSTM models on UNSW-NB15. Form this analysis, it is proved that the proposed MVN-ConvLSTM increases the accuracy performance than on other existing models of identifying intrusions in network system. The proposed system solves the uncertainty issues effectively, because the reason, the accuracy of all three datasets are increased.

4.2 Precision

It is the proportion of positively expected intrusion to all predicted intrusion samples. The Eq. (48) indicates the precision formula.

$$Precision = \frac{TP}{(TP+FP)} \quad (48)$$

Fig. 5 depicts the precision results (%) of the proposed and existing IDS for different dataset. It indicates that the precision of MVN-ConvLSTM is 3.79%, 3.70%, 1.82%, 1.65% and 0.16% higher than Conv-LSTM, SDAE-ELM, DNN-LSTM, CNN-RSA and CNN-LSTM models on CIC-IDS2018. Similarly, the precision of MVN-ConvLSTM is 4%, 3.58%, 2.99%, 1.77% and 0.49% higher than Conv LSTM, SDAE-ELM, DNN-LSTM, CNN-RSA and CNN-LSTM models on WSN-DS. The precision of MVN-ConvLSTM is 1.79%, 1.32%, 0.88%, 0.37% and

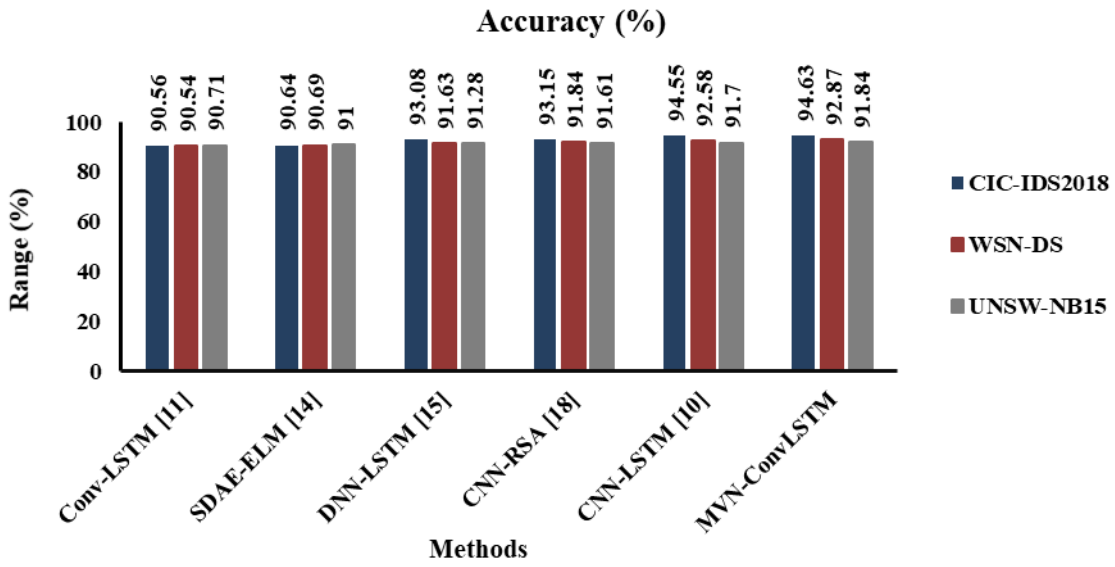

Figure. 4 Accuracy comparison for proposed and existing models on different dataset

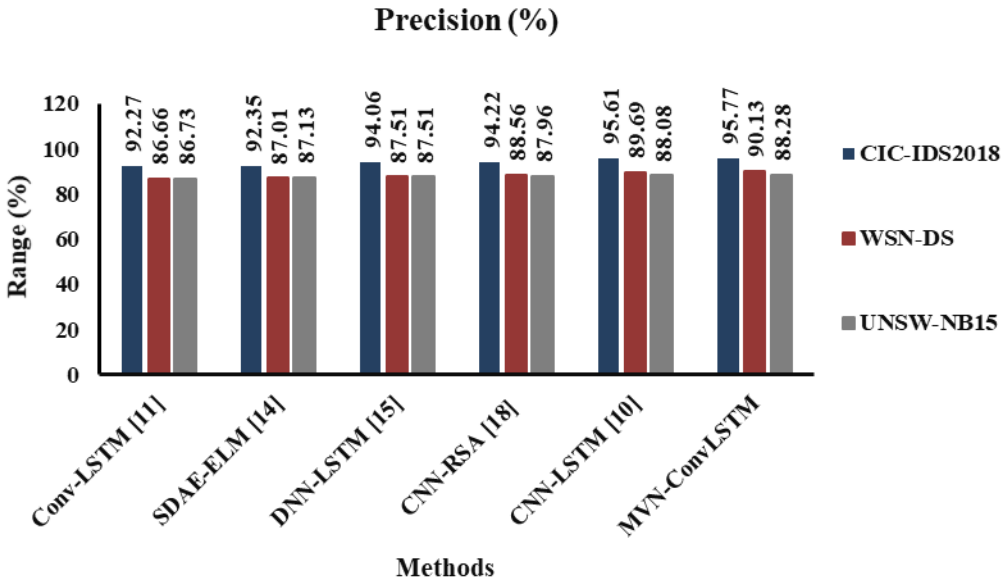

Figure. 5 Analysing the precision of existing and proposed across several dataset

0.23% higher than Conv-LSTM, SDAE-ELM, DNN-LSTM, CNN-RSA and CNN-LSTM models on UNSW-NB15. This analysis shows that the MVN-ConvLSTM improves precision over other models Conv-LSTM, SDAE-ELM, GAN-DNN-LSTM [15], CNN RSA [10] and CNN-LSTM for detecting intrusions. The increasing of TP and reducing of FP in the proposed model increase the overall precision for all three datasets. NS domain reduces the overlapping between membership values of each features for attack and normal domain.

4.3 Recall

Recall is the proportion of confirmed incursion samples to total possible samples which are actually an intrusion. The Eq. (49) depicts the recall formula

$$Recall = \frac{TP}{TP+FN} \tag{49}$$

The recall results (%) of the proposed and existing IDS for different dataset is illustrated in Fig. 6. It indicates that the recall of MVN-ConvLSTM is 2.03%, 1.90%, 1.19%, 1.03% and 1.01% higher than Conv-LSTM, SDAE-ELM, DNN-LSTM, CNN-RSA and CNN-LSTM models on CIC-IDS2018. Similarly,

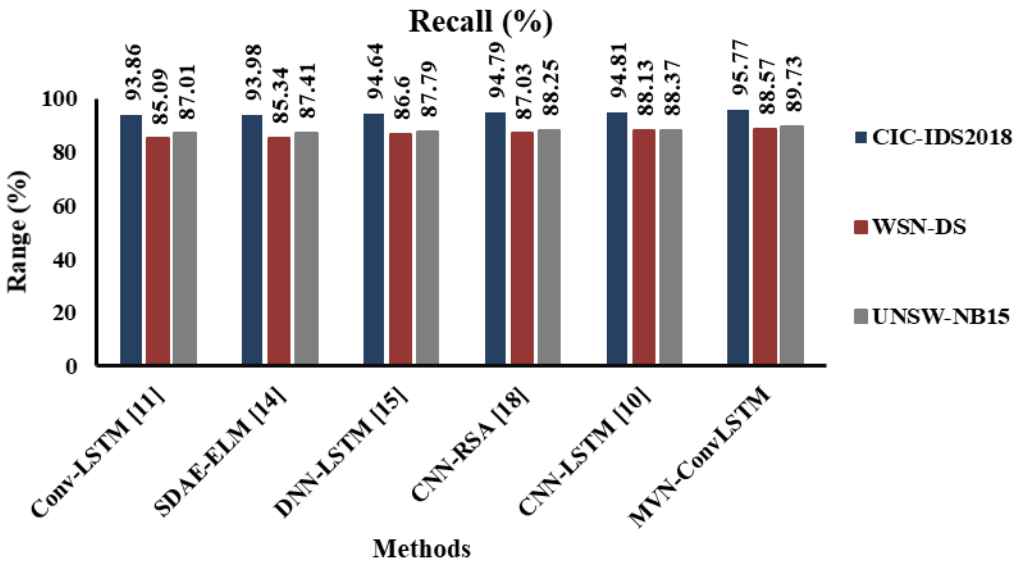

Figure. 6 Recall values for proposed and current models on different datasets

the recall of MVN-ConvLSTM is 4.09%, 3.78%, 2.27%, 1.77%, and 0.49% than Conv-LSTM, SDAE-ELM, DNN-LSTM, CNN-RSA and CNN-LSTM models on WSN-DS. The recall of MVN-ConvLSTM is 3.13%, 2.65%, 2.21%, 1.67%and 1.54% higher than Conv-LSTM, SDAE-ELM, DNN-LSTM, CNN-RSA and CNN-LSTM models on UNSW-NB15. This analysis demonstrates that the proposed MVN-ConvLSTM outperforms other models.

5. Conclusion

In this paper, MVN-ConvLSTM has been developed for the efficient prediction of intrusions. The NS domain is utilised to handle extracted uncertainty features. Four different routes are built using MVN-ConvLSTM for I_A, I_B , T_A and T_B NS domain to obtain a desired result. Finally, the test results on three different datasets (CIC-IDS2018, WSN-DS, and UMSW-NB15) show that the MVN-ConvLSTM obtains the accuracy of 94.63%, 92.84% and 91.84% for three datasets. The accurcay of MVN-ConvLSTM is 17.76%, 12.57%, 8.49%, 7.56% and 3.86% greater than other models mentioned in section 4 on CIC-IDS2018. Similarly, the accuracy of MVN-ConvLSTM is 18.22%, 12.35%, 8.04%, 7.56% and 4.45% greater than other models on WSN-DS. The accuracy of MVN-ConvLSTM is 18.61%, 12.81%, 9.46%, 7.43% and 4.52% other models on UNSW-NB15. In future, an algorithm will be developed to prioritize the feature's importance ranking and that will be added as a layer of CNN model to reduce the complexities of the IDS. The Softmax classifier in MVN-ConvLSTM can also be modified for giving importance for highly ranked features to improve the classifcation accuracy further.

Conflict of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, methodology, software development and validation, Veni; formal analysis, investigation, result checking Sridevi; resources, data curation, writing—original draft preparation, Veni; writing—review and editing, Veni; visualization, supervision, Sridevi.

References

[1] P. Sun, P. Liu, Q. Li, C. Liu, X. Lu, R. Hao and J. Chen, "DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system", *Security and Communication Networks*, pp. 1-11, 2020.

[2] H. J. Liao, C. H. R. Lin, Y. C. Lin and K. Tung, "Intrusion detection system: A comprehensive review", *Journal of Network and Computer Applications*, Vol. 36, No. 1, pp. 16-24, 2013.

[3] H. Alkahtani, and T. H. Aldhyani, "Intrusion detection system to advance internet of things infrastructure-based deep learning algorithms", *Complexity*, pp. 1-18, 2021.

[4] L. Ashiku and C. Dagli, "Network intrusion detection system using deep learning", *Procedia Computer Science*, Vol. 185, pp. 239-247, 2021.

[5] P. Wu, "Deep learning for network intrusion detection:", *Attack Recognition with*

Computational Intelligence (Doctoral Dissertation, UNSW Sydney), 2020.
[6] K. Rajasekaran and K. Nirmala, "Classification and importance of intrusion detection system", *International Journal of Computer Science and Information Security*, Vol. 10, No. 8, p. 44, 2012.
[7] A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges", *Cybersecurity*, Vol. 4, pp. 1-27, 2021.
[8] Z. Wang, "Deep learning-based intrusion detection with adversaries", *IEEE Access*, Vol. 6, pp. 38367-38384, 2018.
[9] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study", *Journal of Information Security and Applications*, Vol. 50, No. 102419, 2020.
[10] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "CNN-LSTM: hybrid deep neural network for network intrusion detection system", *IEEE Access*, Vol. 10, pp. 99837-99849, 2022.
[11] Z. Fan and Z. Cao, "Method of Network Intrusion Discovery Based on Convolutional Long-Short Term Memory Network and Implementation in VSS", *IEEE Access*, Vol. 9, pp. 122744-122753, 2021.
[12] L. Nie, Y. Wu, X. Wang, L. Guo, G. Wang, X. Gao, and S. Li, "Intrusion detection for secure social internet of things based on collaborative edge computing: A generative adversarial network-based approach", *IEEE Transactions on Computational Social Systems*, Vol. 9, No. 1, pp. 134-145, 2021.
[13] N. Anjum, Z. Latif, C. Lee, I. A. Shoukat, and U. Iqbal, "MIND: A Multi-Source Data Fusion Scheme for Intrusion Detection in Networks", *Sensors*, Vol. 21, No.14, p. 4941, 2021.
[14] Z. Wang, Y. Liu, D. He, and S. Chan, "Intrusion detection methods based on integrated deep learning model", *Computers & Security*, Vol. 103, p. 102177, 2021.
[15] T. Kim, and W. Pak, "Early Detection of Network Intrusions Using a GAN-Based One-Class Classifier", *IEEE Access*, Vol. 10, pp. 119357-119367, 2022.
[16] S. P. Thirimanne, L. Jayawardana, L. Yasakethu, P. Liyanaarachchi, and C. Hewage, "Deep neural network based real-time intrusion detection system", *SN Computer Science*, Vol. 3, No. 2, p. 145, 2022.
[17] S. Sheikhi and P. Kostakos, "A Novel Anomaly-Based Intrusion Detection Model Using PSOGWO-Optimized BP Neural Network and GA-Based Feature Selection", *Sensors*, Vol. 22, No. 23, p. 9318, 2022.
[18] A. Dahou, M. A. Elaziz, S. A. Chelloug, M. A. Awadallah, M. A. A. Betar, M. A. A. Qaness, and A. Forestiero, "Intrusion Detection System for IoT Based on Deep Learning and Modified Reptile Search Algorithm", *Computational Intelligence and Neuroscience*, 2022.
[19] V. Torra, "Hesitant fuzzy sets", *International Journal of Intelligent Systems*, Vol. 25, No. 6, pp. 529-539, 2010.
[20] H. Wang, F. Smarandache, Y. Zhang, and R. Sunderraman, "Single valued neutrosophic sets", *Infinite Study*, Vol. 12, 2010.
[21] F. G. Lupiáñez, "Interval neutrosophic sets and topology", *Kybernetes*, Vol. 38, No. 3/4, pp. 621-624, 2009.
[22] B. Zhu, Z. Xu, and M. Xia, "Dual hesitant fuzzy sets" *Journal of Applied mathematics*, 2012.
[23] https://www.kaggle.com/datasets/solarmainframe/ids-intrusion-csv
[24] https://www.kaggle.com/datasets/bassamkasasbeh1/wsnds
[25] https://research.unsw.edu.au/projects/unsw-nb15-dataset