Journal of Autonomous Intelligence (2024) Volume 7 Issue 4
doi: 10.32629/jai.v7i4.1094

ORIGINAL RESEARCH ARTICLE

Detection of Data imbalance in MANET network based on ADSY-AEAMBi-LSTM with DBO Feature selection

Venkatasubramanian Srinivasan[1,*], Vijilius Helena Raj[2], Arunadevi Thirumalraj[3], Kavitha Nagarathinam[4]

[1] *Department of Computer Science and Business Systems, Saranathan College of Engineering, Trichy 620012, India*

[2] *Department of Mathematics, New Horizon College of Engineering, Bengaluru 560103, India*

[3] *Department of Computer Science and Engineering, K. Rama Krishnan College of Technology, Trichy 620015, India*

[4] *Department of Computer Science and Engineering, Mepco Schlenk Engineering College, Sivakasi 62600, India*

*** Corresponding author:** Venkatasubramanian Srinivasan, veeyes@saranathan.ac.in

ABSTRACT

A Mobile Ad Hoc Network (MANET) is a temporary wireless network formed by mobile nodes. These nodes cooperate to relay information in a multi-hop fashion, but some malicious nodes can disrupt the network by providing false routing information. Traditional firewalls and encryption methods can't keep up with the increasing diversity of network threats. To address these issues, Intrusion Detection Systems (IDS) have been developed. In this paper, a new intrusion detection framework named ADSY-AEAMBi-LSTM is introduced. This acronym stands for a bidirectional Long Short-Term Memory (LSTM) model and an adaptive synthetic auto-encoder attention mechanism. The Dung Beetle Optimizer is used to identify optimal features after data preprocessing, followed by classification using ADSY-AEAMBi-LSTM. The study evaluates this model using three datasets: CIC-IDS 2017, UNSW-NB15, and WSN-DS.

Keywords: mobile ad-hoc network; intrusion detection; dung beetle optimizer; attention mechanism; bidirectional long short-term memory

ARTICLE INFO

Received: 5 August 2023
Accepted: 14 November 2023
Available online: 22 January 2024

1. Introduction

Network security has grown in importance as a result of the quick growth in technology and data, incorporating big data analysis, cloud computing, the Internet of Things (IoT), and the growing reliance of everyday life on linked services. The network as a whole will be affected by any vulnerability or danger[1]. Security experts have also realized how crucial it is to create effective networks using intrusion detection systems (IDS) in order to create safe networks. By avoiding unauthorized access, protecting the network's information and communication systems, and ensuring dependability for data carried via computer networks, intrusion detection systems operate to ensure accessibility, safety, and dependability[2], and most importantly accurately and with a minimal number of false alarms in identifying both known and unidentified threats and attacks[3]. Techniques for spotting abuse and abnormalities are used by the intrusion detection system. The first line of defense, often referred to base on previously identified and recorded assaults and threats, a technique known as signature-based identification or abuse detection is used. Comparing this method to previous models, it identifies a greater variety of incidences and has a lower percentage of false alarms. The approach is

vulnerable to such assaults because attackers are developing novel and previously unheard-oftechniques as networks and services grow[4].

A wide range of mobile nodes make up the MANET that may join to the network depending on demand. The MANET system uses each network packet as a router[5]. The two components that make up MANET's functioning are data blockage and communication environment time. With its many social networking setups, the MANET network provides a useful base. The conveyance of information based on response and inquiry is a component of data transmission in the network[6]. MANET is generally prone to a number of security problems, including routing table overflow, wormhole, packet replication, poisoning, spying, or denial of service (DoS). Attacks stop packets from getting to their final destination by interrupting the flow of data between nodes[7]. Few mobile phones utilized as WIFI interfaces in MANET create a moderate correlation without the aid of any organized infrastructure or centralized administration. Hosts reachable by all of them must depend on a different computer to serve as the email relay for the site, even if the hosting spaces for each particular cellular node intrusion may differ[8]. The problem with MANETs is that they are dynamic, which makes it harder for them to be renowned for offering comprehensive answers to security- or QoS-related problems[9]. The nodes in a MANET act as a conduit for information transmission from the point of origin to the destination if every node is located within the defined area or border[10].

Artificial neural networks with several hidden layers are used in conjunction with deep learning, a branch of artificial intelligence[3]. Deep learning has a number of performance attributes that make it appropriate for the creation of an IDS, such as the scalability, adaptability, and dependability of the algorithms employed in DL[11]. Deep learning (DL) aids the detection of system faults and vulnerabilities by security specialists by assisting in data summary and visualization. Many methods based on deep learning (DL) have been used to improve detection rates and flexibility in the intrusion detection sector[12]. There have been many deep learning-based IDS approaches developed recently[13].

The contributions given in this work are as follows:

- To improve the detection efficiency, DBO optimization is used for feature selection to select the optimal feature subset.
- This ADSY-AEAMBi-LSTM model can classify MANET precise traffic data;
- In order to address the issue of uneven network data, ADSY is utilized to augment the data from samples belonging to the minority category, finally resulting in a roughly symmetrical distribution The amount of each sample type there are overall, enabling the model to train well;
- In order to increase data fusion, a better stacked autoencoder (SAE)is created and used to reduce the dimensionality of data.
- This study uses the UNSW-NB15, CIC-IDS 2017, and WSN-DS datasets for simulation evaluation to train and evaluate the ADSY-AEAMBi-LSTM model's effectiveness.

The remaining sections of the study are organized in the form of shadows: The relevant works are summarized in Section 2, the suggested model is briefly explained in Section 3, Section 4 presents the findings and the validation analysis, while Section 5 offers a summary and a conclusion.

2. Related works

The major emphasis of the Deep Learning-based IDS presented by Meddeb et al.[14] in labelled datasets used for intrusion detection was the Denial of Services (DoS) attacks. The functionality of routing in mobile networks may be affected by a wide range of potential assaults. The Stacked AE-IDS approach improves IDS's capacity to detect assaults in MANETs by lowering coupling and simulating a high-level overview of pertinent components. This strategy is essential for MANET security since it particularly addresses DoS occurrences and how they affect mobile networks' ability to route traffic. The Stacking AE-IDS method has the ability to

improve the safety of MANETs and boost the efficiency of IDSs by distinguishing various attacks, particularly DoS attacks, and comprehending their effects on the route services provided by mobile networks.

In this paper, Prashanth et al.[15] proposed a system that integrates optimization and classification methodologies for accurately predicting classified labels. The system also includes feature extraction, optimization, and classification in addition to preliminary processing. Addressing duplicate data and cases of missing values is part of the first preparation of the provided data. Following that, a set of traits are picked using the Principal Component Analysis (PCA) method that enhance classification performance. The IDS aspects that are most important are chosen using the Grey Wolf Optimization (GWO) approach. This approach makes IDS more approachable overall. In order to determine whether an attack or a usual outcome will occur, the Deterministic Convolutional Neural Network (DCNN) technique is used, based on the classification results. The performance of the suggested framework is evaluated using a range of measures, and its outcomes are contrasted with the results from the most recent cutting-edge models.

An article by Prasad et al.[16] featured network configuration, data production, feature extraction, an intrusion detection technique, sample labelling, and a model for evaluating the dependability and effectiveness. The evaluation model, which applies a fuzzy logic system, evaluates the efficacy of intrusion detection, the robustness of its hardware systems, as well as the effectiveness and dependability of different techniques. The results demonstrate how several statistical performance indicators are traded off as a result of an unbalanced sample ratio. The recommended assessment method rates the scheme dependability of a system for detecting intrusions on the two best results. According to experimental findings. The suggested detection method works better at sustaining high scheme dependability than current methods.

The major finding of this research by Ponnusamy et al.[17]is the lack of communication traces that may be utilised to train the existing machine learning algorithms to differentiate IoT-specific breaches. We specifically examine the Knowledge Discovery and Discovery in Databases (KDD) Tournament database to illustrate the design difficulties of wireless detection of intrusions based on current data properties. There are several suggestions offered to improve a wireless network's traffic capturing techniques future-proof. The study paper’s introduction examines various placement tactics, approaches to data gathering, and techniques for detecting intrusions. Investigating the design challenges involved in creating an IDS in a wireless context is the main goal of this effort. It is more difficult to create an intrusion detection system Compared to a wired network, a wireless network is more convenient environment due to the complexity of the architectural architecture. Thus, in addition to discussing future wireless services and design difficulties in the context of cellular networks, this paper also discusses the fundamental wired detection and deployment methodologies. The three main wireless environments to focus on are the Internet of Things (IoT), mobile ad hoc networks, or the use of wireless sensor networks (WSN), as they represent future developments and are frequently the subject of assaults. Consequently, it is essential to develop an IDS that focuses on wireless networks.

Sbai and Elboukhari[18] suggested using a knowledge-based intrusion detection system (KBIDS) to defend MANETs against SYN flooding and UDP/data DDoS attacks, two types of DDoS attacks. Utilising the CICDDoS2019 dataset, the DL precise DNN method is applied. The results of simulation studies indicate that the proposed architectural paradigm may provide results and performance metrics (Accuracy, precision, Recall, and F1-score) that are quite fascinating and interesting.

Through efficiency evaluation, malicious node identification, and network attack mitigation, the authors of this study, Abbood et al[19], focused on security standards. By using the three techniques—Cascading-Back-Propagation-Neural Network (CBPNN), Feedforward-Neural Network (FNN), and CBPNN (FFNN)complex patterns in MANET were discovered. The effectiveness of intrusion detection systems (IDS) and how well they work with machine learning (ML) are often improved by the use of convolutional neural networks (CNN) and these essential DNN building elements. Compared to its logical and statistical competitors, machine

learning (ML) methodologies are superior in MANET network training and facilitating adaptability to various environments. End-to-end (E2E) and average receiving packet (ARP) performance characteristics show that the suggested model performs better than a different current model.

An Exponential-Henry Gas Solubility Optimisation (EHGSO)-based intrusion detection method for MANET is the research suggested by Ninu[20]. The newly created EHGSO algorithm is used to select the most optimal routes early in the secure routing process. The fitness variables for this technique include power, neighbour value, distance, and connection quality. The Henry Gas Solubility Optimisation (HGSO) and the Exponential Weighted Moving Average (EWMA) are both included in the proposed EHGSO. The second phase, in which the transmitted data packets are altered and Knowledge discovery in databases (KDD) attributes are retrieved, starts the intrusion detection phase on the base station. Following the extraction of the KDD features, data augmentation is performed. Before performing intrusion detection, a Deep Neuro Fuzzy Network is trained using the proposed EHGSO method. The suggested method performs better than all already in use technologies. The recommended approach generates the following values in the absence of attacks: 4.123, 0.086, 95.877%, 0.342 J, 134975 kbps, 0.950, and 0.924. These measures include jitter, recall, accuracy, & packet corruption as well as Performance and Developmental Review (PDR).

3. Proposed methods

ADSY-AEAMBi-LSTM is used in this study's construction of a system for intrusion detection. **Figure 1** shows the technique of the ADSY-AEAMBi-LSTM based IDS model.

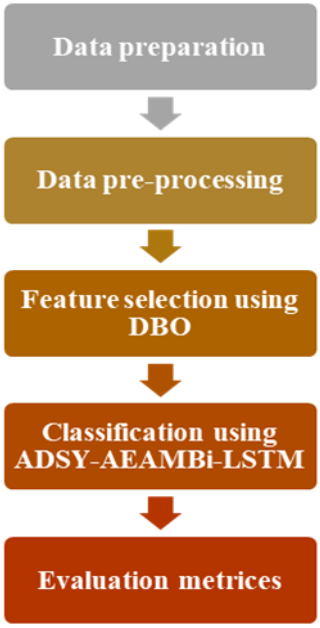

Figure 1. Flow of the work.

3.1. Dataset description

In order to build an ADSY-AEAMBi-LSTM IDS, the right datasets must first be chosen. In order to accurately reflect what the model would see in the actual world, the dataset should contain both legitimate and malicious entries. Our study makes use of the recently made available UNSW-NB15, CIC-IDS2017, and WSN-DS datasets. These datasets, which are considered to be fresh and do not include a considerable amount of redundant information, contain data on both legitimate and illicit traffic.

3.1.1. CIC-IDS2017

The CIC-IDS2017 dataset, which stands for the "Canadian Institute for Cybersecurity Intrusion Detection Dataset 2017" is a comprehensive dataset used for research and development in the field of network security and intrusion detection. It was created by the Canadian Institute for cyber security and is designed to assist in the evaluation and benchmarking of intrusion detection systems and techniques. The dataset contains a wide range of network traffic data, including both benign and malicious network activities. Eleven additional attacks are covered by CIC-IDS2017, including FTP-Patator and SSH-Patator as well as Brute Force, PortScan, DoS, and online assaults like XSS and SQL Injection. It was established in 2017 by the Canadian Institute of cyber security and its 80 characteristics are used to track both good and bad traffic[21].

3.1.2. UNSW-NB15

The UNSW-NB15 dataset, which stands for "University of New South Wales Network-Based 2015" is a widely used network traffic dataset created by the University of New South Wales (UNSW) for the purpose of research and development in the field of network security and intrusion detection. This dataset is designed to evaluate intrusion detection systems and test various algorithms and models. Data on nine different kinds of assaults, comprising fuzzers, analysis, covert access points, denial-of-service attacks, exploits, etc, are included in this dataset. Additionally, it has logs of secure traffic. The Australian Centre for Cyber Security (ACCS) founded it in 2015. Information was collected on trustworthy sources as Microsoft Security Bulletin, Symantec Corp's BID, and CVE's prevalent vulnerabilities and exposures[21].

3.1.3. WSN-DS

In 2016, WSN-DS was established to maintain track of the number of sensor-equipped wireless device nodes in networks in order to distinguish between legal and harmful traffic. The records from this collection, which are characterized by 23 attributes, are extracted using the LEACH routing algorithm. In addition to regular records, there are four other forms of DoS assaults flooding, including Grayhole, Blackhole, and TDMA[22].

3.2. Data pre-processing

Figure 2. Steps for Preprocessing.

3.2.1. Load datasets

It is possible to get free access to the datasets utilized in this inquiry. A CSV file in pcap formats is used to store the data. Each dataset's specifics have been downloaded using the Pandas package in this phase, and each dataset's specifics were then cleansed of any null or duplicate information to prepare it for the next stage.

3.2.2. Encoding of data

This dataset provides data on nine unique forms of attacks, like fuzzers, evaluation, hidden doors, denial-of-service attacks, exploits, etc. It also maintains data of safe traffic. It was introduced in 2015 by the Australian Centre over Cyber Security (ACCS). Information was obtained from BID (Symantec Corp), the trustworthy websites CVE (Common Risks and Exposures), & MSD (Microsoft Corp) (including the Microsoft Security Bulletin)[21].The YOLO feature extractor is typically a deep convolutional neural network (CNN) that processes the input image and produces a feature map. This feature map contains spatial information about the objects in the image, which is later used for bounding box prediction and classifying the objects. The feature extractor is designed to capture various levels of detail and abstraction in the image, making it capable of detecting objects of different sizes and complexities.

3.2.3. Normalization of data

To enhance within-range qualities, normalization of the data is a preprocessing method utilized. The degree to which students learn will depend on the data's variation they get from the CSV file, that contains a lot of common derivations and means. Standard Scalar was used to scale the data used as input for this study, producing results with a mean and standard deviation of 0 and 1, respectively. The datasets have been normalized using Standard Scalar, a collection of tools for "sklearn. preprocessing".

3.2.4. Data splitting

The dataset for the model was split into training and testing sets. This study also separates the training set in sets for training and validation to optimize our the hyperparameters while training and improve model performance. The size of the two sets was chosen cross-validation with the strategic K-Fold method procedure depending on the value of K.

3.3. Feature selection using DBO

The Dung Beetle Optimizer (DBO) is an innovative algorithm that leverages the actions of beetles such as rolling balls, dancing, scavenging, thieving, reproducing, and engaging in other behaviors. This method is distinguished by its quick convergence and strong desire for excellence. The DBO algorithm consists of four primary stages: ball rolling, reproduction, food exploration, and theft. When a ball is unobstructed, the goal of rolling is to enhance the where the dung beetle is located. This is based on the notion that the posture of a dung beetle is affected by the intensity of light.

$$x_i(t+1) = x_i(t) + \alpha.k.x_i(t-1) + b.\Delta x \quad (1)$$
$$\Delta x = |x_i(t) - X^w|$$

where in *t* is the present number of iterations, $x_i(t)$ denotes the location details for the i-th iteration's preying mantis, and $k \in$ (0,0.2] suggests a fixed amount of the displacement coefficient and the at present repetition count, The variable that was allocated to (0,1) is represented by the integer b and αreflects a natural factor given to a value of -1 or 1,$X - w.X^w$is where the ball is at its lowest point, and Δx is employed to simulate variations in the intensity of light[23].

The dung beetle changes its trajectory to go another direction when it encounters an impenetrable obstacle. The approach represents the dance motions using a tangent function. The dung beetle's location has changed as a result of the ball's altered direction and continued motion.

$$x_i(t+1) = x_i(t) + \tan(\theta)\,|x_i(t) - x_i(t-1)| \quad (2)$$

The shape of a algorithm mimics the same scarab spawning region as Equation 3 by using an edge selection mechanism during reproduction.

$$\begin{cases} Lb^* = \max(X^* .(1-R), Lb) \\ Ub^* = \min(X^* .(1-R), Ub) \end{cases} \quad (3)$$

where, X^*depicts the most effective solution at this time , while Lb^* represents the optimal solution, and Ub^*reflects the best possible outcome of the best possible outcome. R = 1-$\frac{t}{T}$ and T is the most repetitions possible, the ideal solution's upper and lower limits are represented by Lb, and the highest possible level of solution is represented by Ub[23].

The dung beetle only lays a single egg per iteration once the egg-laying zone has been identified. It is evident from (12) because the placement of l eggs is dynamically changed during the iteration, as does the egg-laying area as Equation 4[23].

$$B_i(t-1) = X^* + b_1 .(B_i(t) - Lb^*) + b_2 .(B_i(t) - Ub^*) \quad (4)$$

Where, $B_i(t)$ is a location at the t-th iteration in the i-th sphere, D is the dimensions of the ideal solution, while b_1 and b_2 are two separate randomised vectors of size 1 × D, respectively[23].

The perimeter of the ideal predation region is established during the devouring process based on the movements of the insect's positions.

$$\begin{cases} Lb^b = \max\left(X^b . (1-R), Lb\right) \\ Ub^b = \min\left(X^b . (1+R), Ub\right) \end{cases} \tag{5}$$

Where, X^b is worldwide optimization, Lb^b is the ideal searching domain's limit is lower, and Ub^b is the highest point in the ideal searching domain. The following is a new update on the location of the tiny beetle.

$$x_i(t+1) = x_i(t) + C_1 . \left(x_i(t) - Lb^b\right) + C_2 . \left(x_i(t) - Ub^b\right) \tag{6}$$

Where, $x_i(t)$ indicates the i-th dung beetle's position information at the twelfth repetition, C_1 An integer at random that complies with the standard distribution, and C_2 denotes a random vector belongs to (0,1)[23].

The following information is updated regarding the dung beetle's location throughout the stealing phase.

$$x_i(t-1) = X^b + S . g . (|x_i(t) - X^*| + |x_i(t) - X^b|) \tag{7}$$

where, $x_i(t)$denotes the i-th thief's position data at the tth iteration, g a 1 ×D random vectors that follows a typical distribution, and S an integer that value[24].

3.4. Classification using ADSY-AEAMBi-LSTM

3.4.1. ADSY

The adaptive oversampling method known as ADSY uses samples from minority classes. By providing more examples in a particular place with less density and a smaller number in a feature a densely populated area, it differs from past data expansion methods. ADSY is superior to other types of managing traffic on networks in MANETs with significant data imbalance using data augmentation methods because it adaptively pushes decision limits to hard-to-learn samples. The following is how the algorithm is applied:

Step 1: Calculate G, which can be stated as the number of samples that need to be synthesized.

$$G = (n_b - n_s) \times \beta \tag{8}$$

where $n_b$consists of a substantial sampling, while n_s stands for the minority samples and β ∈(0, 1). Step 2: Using the Euclidean distance, determine K neighbors for every minority sample, and then specify by $r_i$the percentage of class samples from the majority that are present in the neighbours, which is

$$r_i = {}^{k}/_{K} \tag{9}$$

where k is the majority of the class samples in the most recent neighbor and K represents a current value of the neighbors.

Step 3: Use G to determine the number of samples needed for each minority sample, and then use Equation (11), that may be written as

$$g = G \times r_i \tag{10}$$

$$Z_i = X_i + (X_{Zi} - X_i) \times \lambda \tag{11}$$

where g stands for the quantity that needs to be created, Z_i represents the freshly created sample, X_iThe present minority sample, and X_{Zi} is a representative sample of the k neighbours' random minorities of X_i,λ ϵ(0, 1).

3.4.2. Autoencoder (AE)

AEs are unsupervised learning networks that have an intermediary layer with fewer nodes then the left and right sides with input and output dimensions that are equal. An encoder and a decoder are the two fundamental parts of the conventional AE shown in **Figure 3**. DL techniques are used to discover the right version of what is provided while keeping information. To put it simply, the encoder reduces the original data's dimensions to generate a representation, and the decoder then reconstructs that representation to recover the original data. Under this fundamental idea, it is feasible to utilize the trained encoder to decrease the dimensionality of the data. The AE may make nonlinear modifications, which enables it to learn more thorough projection data information, in contrast to the standard PCA data reduction dimension technique[25].

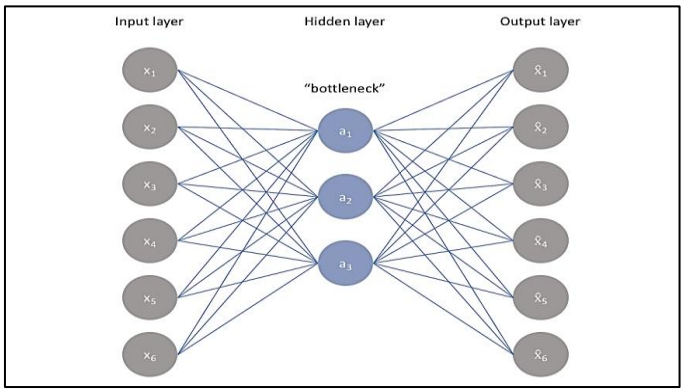

Figure 3. Auto-encoder.

The objective of this article was to present an AE that can decrease data dimensionality while also enhancing data resilience in order to handle complicated MANET network situations. Although the AE may decrease data dimensionality more efficiently than existing dimensionality reduction approaches, this paper also sought to provide an AE that can do both of these things. Throughout network training cycles, dropout enables the removal of each neuron with a probability of p. Consequently, each neuron is less reliant on other neurons. which reduces overfitting and, to some degree, enhances the model's capacity to generalize. After dimensionality reduction, by combining dropping and SAE methods, a low-latitude depiction is created. Each dimension after dimensionality reduction has the potential to be rejected, resulting in a larger information set for each dimension than would be acquired by a typical AE and enabling more efficient model education. Using the ideas stated above as a foundation, we developed an improved multilayer encoder structure shown in **Figure 4**[25].

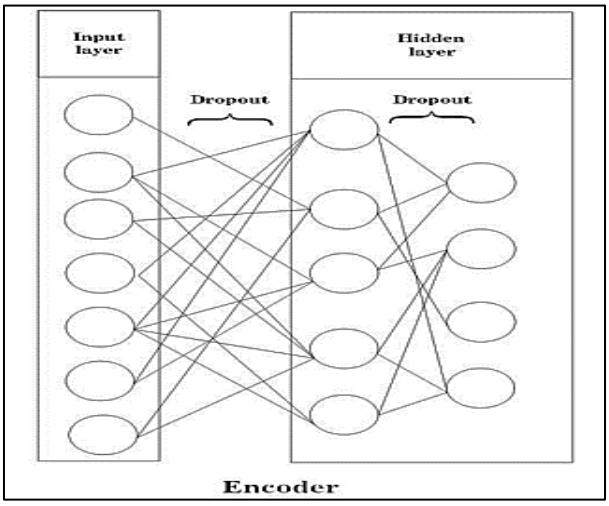

Figure 4. Stacked auto-encoder.

3.4.3. Channel attention

Based on the notion that individuals prefer to concentrate on certain tiny portions of an image rather than the whole picture when they examine it, an attention mechanism was created. With a squeeze-and-excitation (SE) channel-based network AM, the WMW team at ImageNet 2017 won the Image Classification competition. Convolutional block attention module (CBAM) enhancements based on SE include the addition of a Maxpool channel. This might significantly enhance the model's classification ability, as shown by the researcher's several efforts. Our research modified the CBAM utilized in 3D processing of images based on these concepts, and it was subsequently applied to the IDS algorithm for 2D data. The two crucial processes of squeezing and excitation are shown in **Figure 5**[25], which illustrates the CBAM's 2D data processing procedure. To give global information for all channels, traffic data is averaged or maximally pooled from the (c, w)-dimensional format to a (c, 1)-dimensional form during the squeeze phase. During the excitation phase, the compressed data is adaptively transformed using a multilayer perceptron (MLP), resulting in a weighting matrix of values for each and every channel.

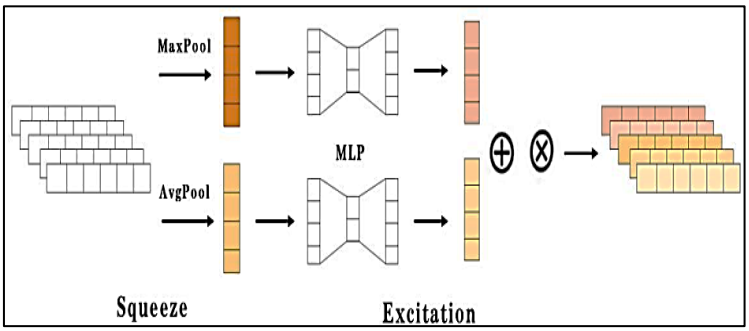

Figure 5. CBAM.

3.4.4. Bidirectional LSTM

In order to solve the recurrent neural network's(RNN) long-term dependence problem, LSTM[26] statuses of storage cells is provided. When RNNs calculate the relationship between distant nodes, a problem called long-term dependence, also known as gradient dispersion or gradient explosion, is created. The LSTM network's single-time step updating process is shown in the diagram below:

$$i_t = \sigma(W_{xi}X_t + W_{hi}h_{t-1} + b_i) \tag{12}$$

$$f_t = \sigma\left(W_{xf}X_t + W_{hf}h_{t-1} + b_f\right) \tag{13}$$

$$o_t = \sigma(W_{xo}X_t + W_{ho}h_{t-1} + b_o) \tag{14}$$

$$\widetilde{c_t} = \sigma(W_{xc}X_t + W_{hc}h_{t-1} + b_c) \tag{15}$$

$$c_t = f_t c_{t-1} + i_t \widetilde{c_t} \tag{16}$$

$$h_t = o_t \tan \mathrm{h}(c_t) \tag{17}$$

where i_t, f_t, and $o_t$input, forget, and output gates should be shown in that sequence. Two different activation functions are represented, respectively, by the symbols0020 σ(sigmoid) and (tanh). $c_t$depicts the status of a cell right now, c_{t-1} indicates the previous state of the cell, and $\widetilde{c_t}$depicts the potential memory cell. h_t depicts the potential memory cell, and h_{t-1} indicates the previous cell's concealed state.

Retroactive hidden states added to the Bi-LSTM network enhance the LSTM$\overleftarrow{h_t}$ to the present concealed forward states $\overrightarrow{h_t}$, making it possible for it to develop a prospective capability similar to the model known as

the hidden Markov model (HMM). The Bi-LSTM network upgrade themselves over just one time step, as demonstrated by the following:

$$\overrightarrow{h_t} = \tan h(W_{h\vec{t}}X_t + W_{\vec{h}\vec{h}}\vec{h}_{t-1} + b_{\vec{h}}) \tag{18}$$

$$\overleftarrow{h_t} = \tan h(W_{h\overleftarrow{t}}X_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}) \tag{19}$$

$$h_t = \overrightarrow{h_t} + \overleftarrow{h_t} \tag{20}$$

where h_t indicates current cell's hidden state, h_{t-1} indicates the preceding cell's concealed state, $\overrightarrow{h_t}$indicates the current cell's concealed forward state, and $\overleftarrow{h_t}$signifies the current cell's hidden reverse state.

The Bi-LSTM's structure is depicted in **Figure 6**[27]. For MANET network traffic, the Bi-LSTM may efficiently make use of the periodic properties included in the contextual data in order to optimize model training.

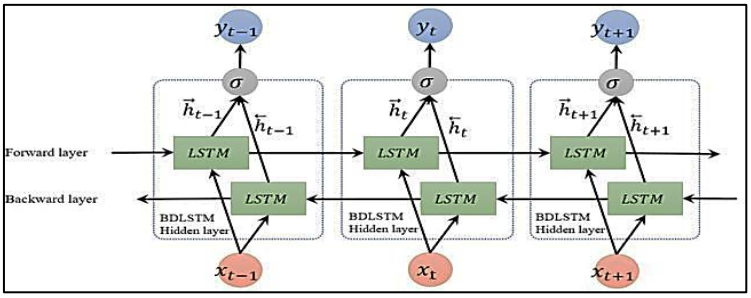

Figure 6. Bi-LSTM structure.

3.4.5. Network architecture

Figure 7 depicts the fundamental architecture of the ADSY-AEAMBi-LSTM model, which comprises of an input layer, an encoding layer, a layer with many convolutions, a layer having attention, a layer with Bi-LSTM, a layer with entirely connected layers, and an output layer. At the lowest layer, the dataset gives communication data to the model. The updated SAE's encoder element, It is applied in the transmitter layer of the model and has been trained effectively to reduce dimensionality on the data. The model applies a variety of convolutional algorithms to the downscaled data across the multi-convolutional layer in order to extract features. At the attention layers the model modifies the weights of each channel and gives streams with higher importance more weight using the CBAM. By gathering data on characteristics for each dimension, the algorithm learns the connections between the parameters in the Bi-LSTM layers. The model's fully connected layer has a classifier that receives the newly discovered features as input and outputs the classification results to the resulting layer. The training of the ADSY-AEAMBi-LSTM model is shown in Algorithm 1.

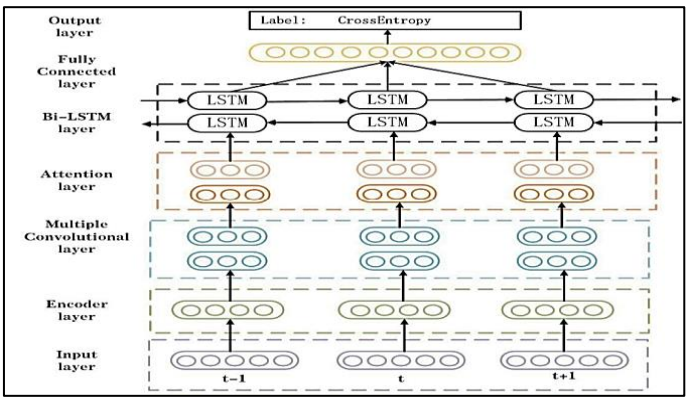

Figure 7. Overall architecture.

Algorithm 1 ADSY-AEAMBi-LSTMTraining

Input: CIC-IDS 2017, UNSW-NB15, and WSN-DS datasets
Output: Accuracy, Precision, DR and FAR
1: **For** data gathered from the training set or the test set;**do**
2: single-shot encoding;
3: If training set,
4: ADSY data augmentation;
5: Normalization;
6: **End**.
7: Feature selection using Dung Beetle Optimizer
8: **For** data taken from a training set or the test set; **do**
9: Reduce data dimensionality by using an encoder;
10: Perform out steps for multilayer convolution;
11: Republish channel weights using CDAM;
12: Obtain sequence data using Bi-LSTM;
13: Make the dimension flat;
14: Classify after sending to the fully linked layer;
15: **End**.
16: Test model on CIC-IDS 2017, UNSW-NB15, and WSN-DS+;
17: Adam will collect loss and update ADSY-AEAMBi-LSTM;
18: return accuracy, precision, DR and FAR.

4. Result and discussion

ADSY-AEAMBi-LSTM was created utilizing a Dell Inspiron 15 3511 computer with an Intel(R) Core(TM) i7-1165G7 CPU operating at 2.80 GHz and 8.00 GB of RAM. Pandas, TensorFlow,& Keras were used in the construction of the deep learning model. Each model was tested using a binary classification strategy. **Table 1** Shows the experimental scenario Binary categorization was performed on the datasets by dividing them into benign and assault categories. The dataset is classified as innocuous or as a single type of assault for binary classification, as shown in **Table 2**.

Evaluation metrics

Table 1 displays the parameters of the confusion matrix are employed to evaluate the efficiency of IDS in MANET. The symbols TP, FP, TN, and FN stand for benign information that were mistakenly categorized as harmful, malicious, benign, and malicious records which were erroneously categorized as benign, respectively. The confusion matrix indicators (FAR) are used in this study to calculate rate of detection (DR), the accuracy (ACC),precision (Pr), & false alarm rate (FAR). The ACC is the percentage of records' accurate predictions that were made. The ability to predict only positively-skewed data as a whole is commonly known as DR. avoiding misunderstanding adverse data as positive is known as Pr, and the proportion of typical traffic incorrect classifications is known as FAR.

Table 1. Experimental scenario.

Dataset	No. of records	Types of records
CIC-IDS 2017	2	Normal and malicious
UNSW-NB15	2	Normal and malicious
WSN-DS	2	Normal and malicious

Table 2. Confusion matrices.

Actual class	Predicted as negative	Predicted as positive
Labelled as negative	TN	FP
Labelled as Positive	FN	TP

FP stands for false positive, FN for false negative, and TN for true negative.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{21}$$

$$DR = \frac{TP}{TP + FN} \tag{22}$$

$$FAR = \frac{FP}{FP + TN} \tag{23}$$

$$Pr = \frac{TP}{TP + FP} \tag{24}$$

In the analysis of CIC-IDS2017 dataset, CNN achieved 97.42% of accuracy, 96.38% of precision, 97.12% of DR and 19% of FAR. LSTM achieved 98.12% of accuracy, 97.45% of precision, 98.41% of DR and 16% of FAR. DBN achieved 98.93% of accuracy, 98.32% of precision, 98.93% of DR and 10% of FAR. The suggested model succeeded 99.60% of accuracy, 99.54% of precision, 99.56% of DR and 6% of FAR. **Table 3** and **Figure 8** shows the numerical and graphical representation of accuracy, precision and DR analysis. **Figure 11** shows the FAR analysis.

Table 3. CIC-IDS2017 binary classification.

CIC-IDS2017 binary classification based on different classifiers				
Classifier	**Accuracy (%)**	**Precision (%)**	**DR (%)**	**FAR(%)**
CNN	97.42	96.38	97.12	19
LSTM	98.12	97.45	98.41	16
DBN	98. 93	98.32	98.93	10
Proposed model	99.60	99.54	99.56	6

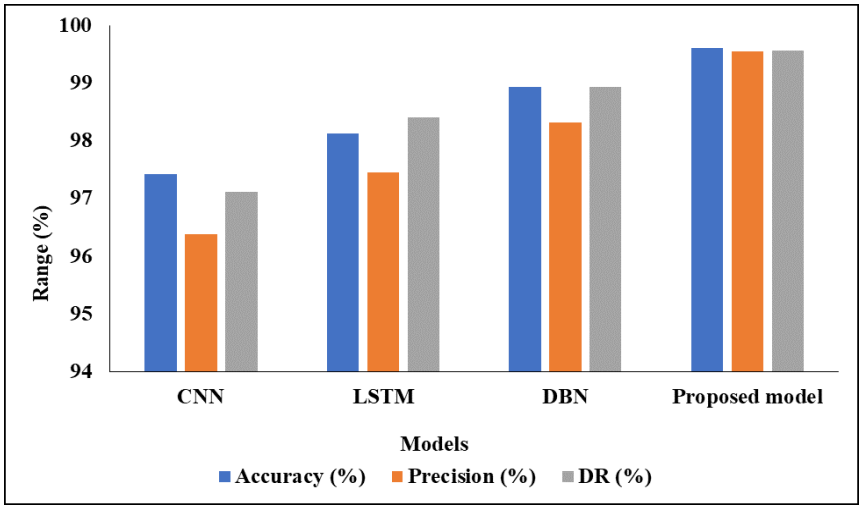

Figure 8. Graphical representation of CIC-IDS2017 dataset.

In the analysis of UNSW-NB15 dataset, CNN achieved 96.34% of accuracy, 96.44% of precision, 96.83% of DR and 18% of FAR. LSTM achieved 97.18% of accuracy, 97.52% of precision, 97.96% of DR and 19% of FAR. DBN achieved 98.62% of accuracy, 98.76% of precision, 98.84% of DR and 9% of FAR. The suggested model succeeded 99.72% of accuracy, 99.45% of precision, 99.32% of DR and 5% of FAR.

Table 4 and **Figure 9** shows the numerical and graphical representation of accuracy, precision and DR analysis.

Table 4. UNSW-NB15 binary classification.

UNSW-NB15 binary classification based on different classifiers				
Classifier	Accuracy (%)	Precision (%)	DR (%)	FAR (%)
CNN	96.34	96.44	96.83	18
LSTM	97.18	97.52	97.96	19
DBN	98. 62	98.76	98.84	9
Proposed model	99.72	99.45	99.32	5

Table 5. WSN-DS binary classification.

Classifier	**Accuracy (%)**	**Precision (%)**	**DR (%)**	**FAR (%)**
CNN	97.72	96.51	97.65	15
LSTM	98.12	97.73	97.88	13
DBN	98. 93	98.23	98.75	8
Proposed model	99.81	99.65	99.23	7

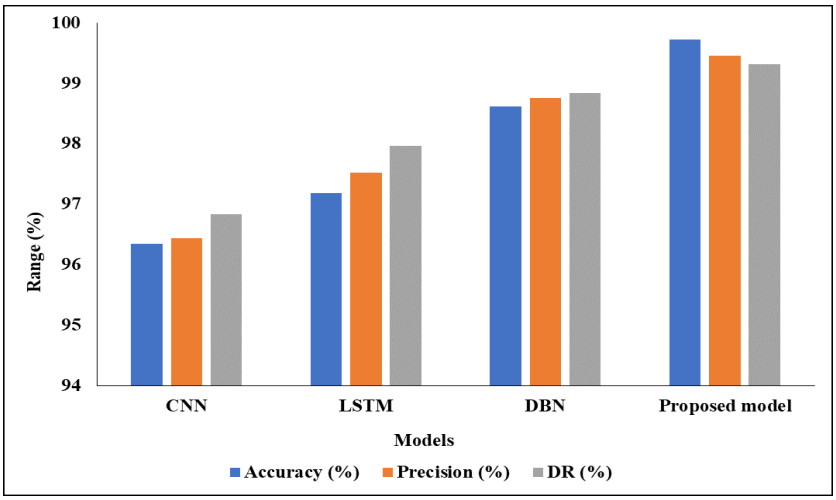

Figure 9. Graphical representation of UNSW-NB15 dataset.

In the analysis of WSN-DS dataset, CNN achieved 97.72% of accuracy, 96.51% of precision, 97.65% of DR and 15% of FAR. LSTM achieved 98.12% of accuracy, 97.73% of precision, 97.88% of DR and 13% of FAR. DBN achieved 98.93% of accuracy, 98.23% of precision, 98.75% of DR and 8% of FAR. The proposed model was successful 99.81% of accuracy, 99.65% of precision, 99.23% of DR and 7% of FAR. **Table 5**, **Figure 10 and Figure 11** shows the numerical and graphical representation of accuracy, precision DR and FAR analysis.

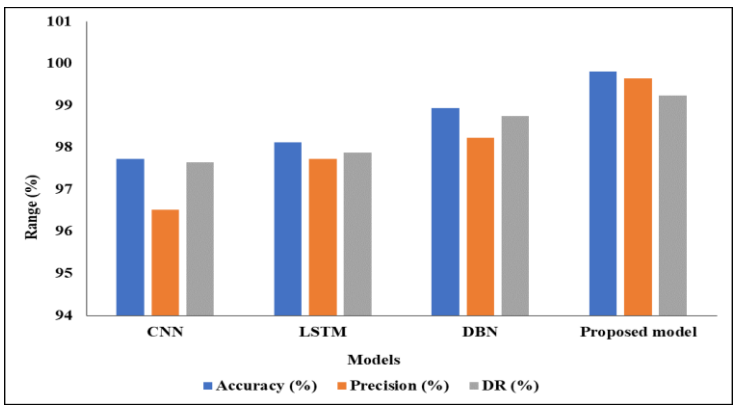

Figure 10. Graphical representation of WSN-DS dataset

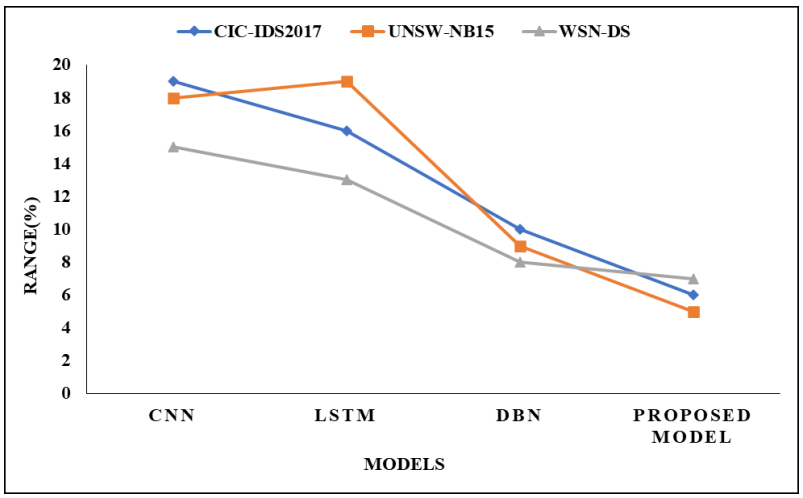

Figure 11. Graphical analysis of FAR.

5. Conclusion

In the end, using DL algorithms to intrusion detection SAE in MANETs is a practical and practicable technique to improve MANET security and reliability. The study proposed an ADSY-AEAMBi-LSTM over sampling algorithm as a data augmentation technique for addressing this network's intrusion data disparity problem, as well as the use of a larger abandonment framework as a data scaling down method enhancing the model's generalization ability, and the network structure and ADSY-AEAMBi-LSTM oversampling methods. Utilizing the data sets UNSW-NB15, WSN-DS, and CIC-IDS2017 the framework was assessed, which comprised both excellent and poor submissions. Despite the model's inability to offer a high detection rate for a variety of threats, including online assaults in CIC-IDS2017, backdoors, and worms, & analyses in UNSW-NB15, the detection rate and FAR scores are good. The feature selection process employs DBO optimization. Compared to other models that are currently available, this model is more accurate. It is believed that the suggested network model is pertinent to the present network IDS development. Future research will concentrate on lowering the model's poor rate of detection and elevated FAR, which are a result of the dataset's inconsistent records. Complexity and Computational Overhead: The ADSY-AEAMBi-LSTM model, especially with feature selection using DBO, can be computationally intensive and complex. This complexity can lead to increased computational overhead, which may not be suitable for resource-constrained MANET nodes.

Author contributions

Conceptualization, VS and VHR; methodology, validation, VS and AT; writing—original draft preparation, review and editing, KN. All authors have read and agreed to the published version of the manuscript.

Conflict of interest

The authors declare no conflict of interest.

References

1. Sun P, Liu P, Li Q, et al. DL-IDS: Extracting Features Using CNN-LSTM Hybrid Network for Intrusion Detection System. Security and Communication Networks. 2020, 2020: 1-11. doi: 10.1155/2020/8890306
2. Alkahtani H, Aldhyani THH. Intrusion Detection System to Advance Internet of Things Infrastructure-Based Deep Learning Algorithms. Uddin MI, ed. Complexity. 2021, 2021: 1-18. doi: 10.1155/2021/5579851
3. Edeh DI. Network intrusion detection system using deep learning technique. M.S. thesis. Dept. Comput, University of Turku, 2021. Turku, Finland. 683..
4. Jabbar MA, Aluvalu R, Reddy S SS. RFAODE: A Novel Ensemble Intrusion Detection System. Procedia

Computer Science. 2017, 115: 226-234. doi: 10.1016/j.procs.2017.09.129
5. Dattatraya KN, Rao KR. Hybrid based cluster head selection for maximizing network lifetime and energy efficiency in WSN. Journal of King Saud University - Computer and Information Sciences. 2022, 34(3): 716-726. doi: 10.1016/j.jksuci.2019.04.003
6. Sharief Shaik M, Mira F. A Comprehensive Mechanism of MANET Network Layer Based Security Attack Prevention. International Journal of Wireless and Microwave Technologies. 2020, 10(1): 38-47. doi: 10.5815/ijwmt.2020.01.04
7. Kantipudi MP, Aluvalu R, Velamuri S. An Intelligent Approach of Intrusion Detection in Mobile Crowd Sourcing Systems in the Context of IoT Based SMART City. Smart Science. 2022, 11(1): 234-240. doi: 10.1080/23080477.2022.2117889
8. Hussain K, Hussain SJ, Jhanjhi N, et al. SYN Flood Attack Detection based on Bayes Estimator (SFADBE) For MANET. 2019 International Conference on Computer and Information Sciences (ICCIS). Published online April 2019. doi: 10.1109/iccisci.2019.8716416
9. N S, Archana KS. Performance Analysis of Machine Learning-based Detection of Sinkhole Network Layer Attack in MANET. International Journal of Advanced Computer Science and Applications. 2022, 13(12). doi: 10.14569/ijacsa.2022.0131262
10. Aluvalu R, Kumaran V. N. S, Thirumalaisamy M, Basheer S, Ali aldhahri E, Selvarajan S. Efficient data transmission on wireless communication through a privacy-enhanced blockchain process. PeerJ Computer Science. 2023, 9: e1308. doi: 10.7717/peerj-cs.1308
11. Benmeziane H. Comparison of deep learning frameworks and compilers. M.S. thesis. 2022. Computer Science Department, École NationaleSuperieured'Informatique. Oued Smar, Algeri.
12. Laqtib S, Yassini KE, Hasnaoui ML. A deep learning methods for intrusion detection systems based machine learning in MANET. Proceedings of the 4th International Conference on Smart City Applications. Published online October 2, 2019. doi: 10.1145/3368756.3369021
13. Venkatasubramanian S. Multistage Optimized Fuzzy Based Intrusion Detection protocol for NIDS in manet. Internation al journal of innovative research in technology. 2021; 8(6): pp.301-311.
14. Meddeb R, Jemili F, Triki B, et al. A Deep Learning based Intrusion Detection Approach for MANET. Published online August 30, 2022. doi: 10.21203/rs.3.rs-1349334/v1
15. Prashanth SK, Iqbal H, Illuri B. An Enhanced Grey Wolf Optimisation–Deterministic Convolutional Neural Network (GWO–DCNN) Model-Based IDS in MANET. Journal of Information & Knowledge Management. 2023, 22(04). doi: 10.1142/s0219649223500107
16. Prasad M, Tripathi S, Dahal K. An intelligent intrusion detection and performance reliability evaluation mechanism in mobile ad-hoc networks. Engineering Applications of Artificial Intelligence. 2023, 119: 105760. doi: 10.1016/j.engappai.2022.105760
17. Ponnusamy V, Humayun M, Z. Jhanjhi N, Yichiet A, Fahhad Almufareh M. Intrusion Detection Systems in Internet of Things and Mobile Ad-Hoc Networks. Computer Systems Science and Engineering. 2022, 40(3): 1199-1215. doi: 10.32604/csse.2022.018518
18. Sbai O, Elboukhari M. Deep learning intrusion detection system for mobile ad hoc networks against flooding attacks. IAES International Journal of Artificial Intelligence (IJ-AI). 2022, 11(3): 878. doi: 10.11591/ijai.v11.i3.pp878-885
19. Ali Abbood Z, Çağdaş Atilla D, Aydin Ç. Intrusion Detection System Through Deep Learning in Routing MANET Networks. Intelligent Automation & Soft Computing. 2023, 37(1): 269-281. doi: 10.32604/iasc.2023.035276
20. Ninu SB. An intrusion detection system using Exponential Henry Gas Solubility Optimization based Deep Neuro Fuzzy Network in MANET. Engineering Applications of Artificial Intelligence. 2023, 123: 105969. doi: 10.1016/j.engappai.2023.105969
21. Halbouni A, Gunawan TS, Habaebi MH, et al. Machine Learning and Deep Learning Approaches for CyberSecurity: A Review. IEEE Access. 2022, 10: 19572-19585. doi: 10.1109/access.2022.3151248
22. Halbouni A, Gunawan TS, Habaebi MH, et al. CNN-LSTM: Hybrid Deep Neural Network for Network Intrusion Detection System. IEEE Access. 2022, 10: 99837-99849. doi: 10.1109/access.2022.3206425
23. Xue J, Shen B. Dung beetle optimizer: a new meta-heuristic algorithm for global optimization. The Journal of Supercomputing. 2022, 79(7): 7305-7336. doi: 10.1007/s11227-022-04959-6
24. Available online:https://www.jeremyjordan.me/content/images/2018/03/Screen-Shot-2018-03-06-at-3.17.13-PM.png (accessed on 2 December 2023).
25. Fu Y, Du Y, Cao Z, et al. A Deep Learning Model for Network Intrusion Detection with Imbalanced Data. Electronics. 2022, 11(6): 898. doi: 10.3390/electronics11060898
26. Gui Z, Sun Y, Yang L, et al. LSI-LSTM: An attention-aware LSTM for real-time driving destination prediction by considering location semantics and location importance of trajectory points. Neurocomputing. 2021, 440: 72-88. doi: 10.1016/j.neucom.2021.01.067
27. Available online:https://ai2-s2-public.s3.amazonaws.com/figures/2017808/f7bdb849dafe17c952bfd88b879e01f74cf59d78/4-Figure3-1.png (accessed on 2 December 2023).