Received: April 18, 2022. Revised: May 30, 2022.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Autoencoder – Support Vector Machine – Grasshopper Optimization for Intrusion Detection System

Sudha Rani Chikkalwar[1,2]* **Yugandhar Garapati[1]**

[1]*Department of Computer science and Engineering, GITAM Deemed to be University, Hyderabad, India*
[2]*Department of Computer science and Informatics, Mahatma Gandhi University, Nalgonda, Telangana, India*
* Corresponding author's Email: yugandhar.garapati@gmail.com

Abstract: An intrusion detection system monitors the networks and identifies the malware or suspicious activity in the network. Machine learning techniques were applied in the Intrusion detection system to improve its efficiency in the identifications. Imbalance data problem in intrusion detection affects the performance of identification and deep learning methods have overfitting problems. The autoencoder – support vector machine – grasshopper optimization (AE-SVM-GO) model is proposed to overcome the limitation of the overfitting problem in intrusion detection. The hybrid technique of AE-SVM-GO is applied to solve imbalance data problem and overfitting problem in intrusion detection. The autoencoder model is applied to generate the instances of minority classes to balance the dataset. The Grasshopper optimization performance hyper-parameter optimization in the SVM model to learn the features to adaptively set the parameter in classification. Four datasets such as UNSW-NB15, CICIDS2017, NSL-KDD, and Kyoto 2006+ dataset were used to test the proposed AE-SVM-GO model. The proposed AE-SVM-GO model has 95.3 % accuracy, whereas the existing convolutional recurrent neural network (CRNN) and SVM-naïve bayes model has 76.82 %, and 93.75 % accuracy respectively.

Keywords: Autoencoder, Grasshopper optimization, Hyper-parameter optimization, Intrusion detection system, Support vector machine.

1. Introduction

A network intrusion detection system is a security method to detect and prevent threat attacks in the network that is facing more challenges. Feature detection based intrusion detection systems were used in the traditional system and this is limited by the refresh rate and scale of a database of predefined signatures. New attack variants were not detected in signature-based intrusion detection systems and detect the known types of attack [1]. Some prevalent security attacks like distributed denial of service (DDoS) causes significant damage to companies. Attack type can be identified based on characteristics extract from the security attacks to prevent the attack in the network. An effective and fast security identification system is required in the network to secure the network [2]. Networks of wireless sensor networks (WSN) and internet of things (IoT) requires data security and need to secure the infrastructure from intrusions. These applications require a high level of security to ensure security and privacy in the network. Simply, an intrusion detection system detects intrusions and abnormal behaviours in the network [3]. There are two types of attacks on the network such as attacks from outside and authorized users seeking greater privileges [4, 5].

Classification based machine learning techniques is growing interest due to their capacity to learn features of malicious behaviours and reduce the number of false alarms. Deep neural networks (DNN) receives increasing attention in a wide range of applications to automatically represents data features and learn hierarchical effectively [6]. Advanced methods such as conventional machine learning systems to handle the detection of small attacks over time. DNN non-linear activation layers facilitate the discovery of effective patterns and maintain effectiveness in drifting conditions [7, 8]. Misuse

detection has poor detection performance in new attacks and anomaly detection have effective performance in detecting unknown attacks that false alarm rate is high [9, 10]. The objectives and contributions of the research is discussed as follows:

1. Autoencoder model is applied to generate minority classes in the dataset to balance the dataset. Autoencoder model generate minority class data instances in the dataset and applied for the classification.
2. Grasshopper optimization technique is applied to optimize the hyper-parameter of the SVM model. The hyper parameter optimization is applied in SVM to increases the learning rate of the classifier.
3. The AE-GO-SVM model has higher efficiency than existing method in intrusion detection technique. The AE-GO-SVM model provides optimal parameter and solves imbalance data problem.

This research paper is organized as follows: The recent research in intrusion detection systems was reviewed in section 2 and AE-SVM-GO was explained in section 3. The simulation setup of AE-SVM-GO is presented in section 4. The result of the proposed AE-SVM-GO is given in section 5, and the conclusion of this research paper is given in section 6.

2. Literature review

Intrusion detection is an important part of network security that involves applying machine learning techniques to identify an abnormality in the network. The recent research in intrusion detection was reviewed in this section.

Gu [11] proposed Naïve Bayes feature embedding and SVM for intrusion detection. Naïve Bayes feature transformation involves generating new high-quality data from original features. Transformed features were used to train SVM for classification. The UNSW-NB15, CICIDS2017, and NSL-KDD datasets were used to test the SVM-Naïve Bayes in the intrusion detection model. The SVM-Naïve Bayes model provides higher performance on multiple datasets and effective performance on identifying the attacks. The model has lower efficiency in handling imbalance datasets and the learning rate of the model is low.

Jiang [12] proposed a hybrid method of one side selection (OSS) and deep hierarchical network to reduce noise samples in major class. The synthetic minority oversampling technique (SMOTE) method was applied to increase the minority class samples and increase the features learning for minority classes. The OSS model established the balanced dataset to improve the learning performance and reduce computational time. The spatial features were extracted by the CNN model and temporal features are extracted by the bidirectional long short term memory (Bi-LSTM) model. The UNSW-NB15 and NSL-KDD datasets were used to evaluate the efficiency of OSS with the deep hierarchical network. The CNN-BiLSTM model creates overfitting in the classification and the model has a limitation of imbalance data problem.

Khan [13] proposed a convolutional recurrent neural network (CRNN) model for the identification of malicious cyberattacks in the network. The local features were captured using convolution in the CNN model and temporal features were captured using the recurrent neural network (RNN) model in the proposed model. The CRNN model combines the benefits of both anomalies based and signature-based intrusion detection models. The CRNN model was evaluated using the CSE-CIC-DS2018 dataset in intrusion detection model and this shows higher efficiency in classification. The spatial and temporal dependencies are captured to solve the problem of exploding and vanishing gradient problems. The CRNN model overcomes the vanishing gradient problem and has limitations in the imbalance data problem.

Andresini [14] combines autoencoders and triplet networks for intrusion detection based on deep metric learning methods. Two separate autoencoders were developed for network attacks and flow in the training stage. Triplet network was trained to learn feature vector embedding to represent network flow. Each flow embedding move was close to reconstruction and autoencoder was restored with the opposite class. Each new flow of predictive stage is related to the class associated with autoencoder for flow reconstruction in the embedding stage. The autoencoder and triplet network have higher efficiency in intrusion detection than existing methods. The overfitting problem in the autoencoder and triplet networks affects the efficiency. The autoencoder and triplet method have a limitation of data imbalance problem in classification.

Choras´ and Pawlicki, [15] applied artificial neural network (ANN) with hyperparameter optimization for the intrusion detection system. The ANN model was tested with CICIDS2017 and NSL-KDD datasets in intrusion detection. The backpropagation was used to improve the learning performance of the ANN model. The tanh, ReLU,

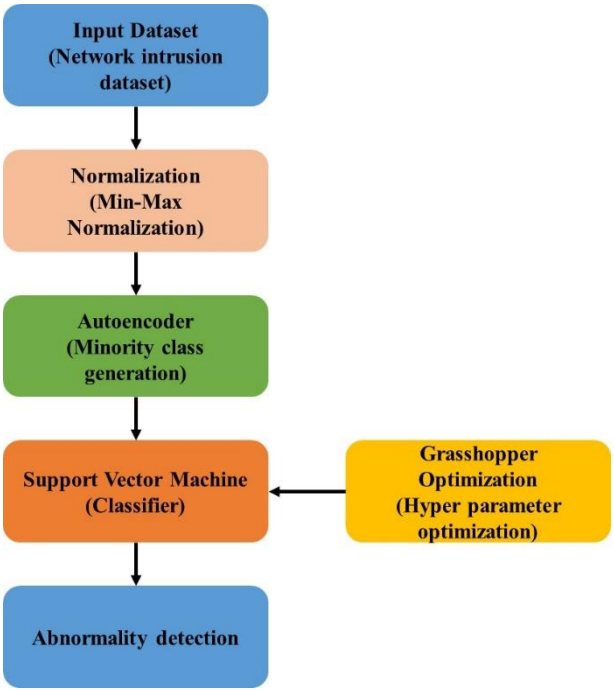

Figure. 1 GO-SVM and autoencoder in intrusion detection

hard sigmoid, and sigmoid activation functions were used as activation functions in the ANN model. The grid search method was applied with various parameters such as neuron nodes, hidden layers, optimizer, activation function, batch size and epoch count. The Adam, rmsprop, and stochastic gradient descent (SGD) optimizers were applied for network optimization. The ANN model has an overfitting problem, and imbalance data problem.

3. Proposed method

Input dataset of network intrusion is applied for min-max normalization and normalization is performed to reduce the difference in input data. Autoencoder is applied to generate the minority class instances that helps to overcome imbalance data. The balanced dataset is applied for training the SVM and hyperparameter of SVM is optimized using Grasshopper optimization method. The GO-SVM and autoencoder model for intrusion detection are shown in Fig. 1.

3.1 Normalization

The normalization method reduces the differences in the input features based on the minimum and maximum values. The classifier model can effectively learn the features due to the less difference in the input data. The min-max normalization formula is given in Eq. (1).

$$x = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

The x_N is normalized data, x is the input value, x_{min} is the minimum value in the feature, and x_{max} is the maximum value in the feature.

3.2 Autoencoder

Stacked auto encoder (SAE) extract high-level features and based on high-level features, the softmax layer performs classification. SAE consists of multi-layer Auto Encoders. Auto Encoder is a deep learning method to measure high-level features to reconstruct input [16–18].

The input layer of the original information is encoded to measure high-level features of the middle layer and the decoding method reconstructs input information. Reconstruction error is minimized by training weights in the network. The x denotes the input data and the hidden layer is measured in Eq. (2).

$$y^i = f(W_1^T x^i + b_1) \quad (2)$$

Where activation function is denoted as $f = \tanh(.)$.

Eq. (3) denotes the decoding of output z.

$$z^i = W_2^T y^i + b_2 \approx x^i \quad (3)$$

Eq. (4) denotes the objective train of autoencoder.

$$J(X, Z) = \frac{1}{2}\sum_{i=1}^{M} \left|\left|x^i - z^i\right|\right|^2 \quad (4)$$

The SAE model is considered a multi-layer autoencoder and SAE is trained based on a layer greedy algorithm. Specifically, the upper layer of the hidden layer vector is used as the input vector of Autoencoder next layer that is pre-training. Network weights of pre-train are connected and fine-tune is applied to obtain the final network weight. SAE represent original features and measure high-level representation.

3.3 Grasshopper optimization – support vector machine

The optimal hyperplane is identified by the SVM model [19, 20] for better generalization of the dataset. This allows a model to predict a new sample for its label and training data set is denoted as $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, where $y\{+1, -1\}$ and $x_i \in R^n$.

The transferred input vector is denoted as x_i and target value is denoted as y_i. Multi-class SVM model

contains the values of class. SVM model develops optimal hyper-plane H from inputs to classify into various classes and Eq. (5) denotes the hyper-plane H.

$$x_i \in R^n : (\vec{w}, \vec{x}) + b = 0, \ \vec{w} \in R^n, \ b \in R \quad (5)$$

Hyper-plane is developed to gives separation of maximum distance between samples of training using Eq. (6).

$$f(\vec{x}) = sign(\vec{w}, \vec{x}) + b \quad (6)$$

SVM of multi-class learning solve it as a binary classification problem and consider it as multiple binary problems. Two linearly separable data are considered for attack traffic detection. Inequality is used to combine optimal hyper-plane is given as in Eq. (7).

$$y_i\{(\vec{w}, \vec{x}) + b\} \geq 1, s.t. i = 1, \dots, n \quad (7)$$

Eq. (8) denotes the optimization problem.

$$y_i = minimization \frac{1}{2}(w^T, w) \quad (8)$$

$$s.t\ y_i(w.x + b) \geq 1$$

Eq. (9) formulate the optimization problem for non-separable case.

$$y_i = minimization \frac{1}{2}(w^T, w) + C \sum_{i=1}^{n} \xi_i \quad (9)$$

$$s.t. y_i(w.x + b) + \xi_i \geq 1; \quad \xi_i \geq 0$$

Where the slack variable is denoted as ξ that helps to select hyper-plane with cost value (C) and less error is the regularization parameter. User empirical investigation is used to obtain optimal C value and Smaller-margin is obtained by a large cost value and this turns it into an overfitting situation.

Mutation probability, cross-over probability, number of iteration, population size, α, and cost value C were parameters used for optimization.

3.3.1. Grasshopper optimization

The grasshoppers optimization algorithm (GOA) method mimics the swarming behaviour of grasshoppers in nature. The possible solution of the optimization problem denotes the grasshopper's position in GOA. Eq. (10) denotes the ith grasshopper position X_i.

$$X_i = S_i + G_i + A_i \quad (10)$$

Where wind advection is denoted as A_i, the ith grasshopper gravity force is denoted as G_i, and the social interaction is denoted as S_i.

Three main components are simulated such as wind advection, gravitational impact, and social interaction. Grasshopper movement of fully simulate components and main components are generated from grasshoppers in social interaction, as given in Eq. (11).

$$S_i = \sum_{j=1}^{N} s(d_{ij}) \widehat{d_{ij}} \quad (11)$$

Where s denotes the social forces strength, d_{ij} are a distance of two grasshoppers, as in Eq. (11) and a unit vector is $d_{ij} = \frac{x_j - x_i}{dij}$.

Social forces are defined in function s, as calculated in Eq. (12).

$$s(r) = fe^{-\frac{r}{l}} - e^{-r} \quad (12)$$

Where l is the attractive length scale and f is attraction intensity.

Repulsion forces are present in the interval of [0, 2.079]. There is no attraction and repulsion, if the distance is equal to 2.079 that is considered as a comfort area. Attraction force increases from 2.079 to nearly 4 and this decreases gradually. The different social behaviours change parameters l and f.

Grasshoppers' interactions related to comfort areas are applied in the model.

Grasshoppers with large distances were not applied with strong forces despite function s merits. The grasshopper is normalized to [1, 4] and mapped to resolve this issue.

Eq. (13) measures G component.

$$G_i = -g\hat{e}_g \quad (13)$$

Where unity vector toward the centre is denoted as $\hat{e}_g$ and gravitational constant is denoted as g.

Eq. (14) denotes A component.

$$A_i = u\hat{e}_w \quad (14)$$

Where wind direction of unity vector is $\hat{e}_w$ and a constant drift is u.

Eq. (15) is developed for grasshoppers with components.

$$X_i = \sum_{j=1}^{N} s(|x_j - x_i|) \frac{x_j - x_i}{d_{ij}} - g\hat{e}_g + u\hat{e}_w \quad (15)$$

Where the number of grasshoppers is N and $s(r) = fe^{-\frac{r}{l}} - e^{-r}$.

A stochastic method is applied to perform exploration and exploitation to solve optimization problems to measure the global optimum of an accurate approximation. Special parameters of the mathematical model are applied in various stages of optimization in exploration and exploitation. A mathematical model is given in Eq. (16).

$$X_i^d = c\left(\sum_{j=1}^{N} c\left(\frac{ub_d - lb_d}{2}\right) s\left(\left|x_j^d - x_i^d\right|\right)\frac{x_j - x_i}{d_{ij}}\right) + \hat{T}_d \quad (16)$$

Where attraction area, repulsion area, and comfort area shrinks to decrease coefficient c, the target value of dth dimension is denoted as $\hat{T}_d$, $s(r) = fe^{-\frac{r}{l}} - e^{-r}$ lower bound in dth dimension is lb_d, and the upper bound is ub_d. Eq. (15) S component is almost similar to S. Gravity is not considered and wind direction is toward a target ($\hat{T}_d$).

As iteration counter increases, the search coverage around target reduces outer c, the number of iterations is proportional to attraction/repulsion forces of grasshoppers in inner c contributes.

Increases exploitation and reduced exploration is carried out with parameter c related to the number of iterations, as shown in Eq. (17).

$$c = cmax - \frac{l(cmax - cmin)}{L} \quad (17)$$

Where L is total iteration, l is current iteration, $cmin$ is minimum value and maximum value is denoted as $cmax$. The $cmax$ and $cmin$ are applied as 1 and 0.00001, respectively.

4. Simulation setup

Intrusion detection dataset details and parameter settings of GO-SVM with autoencoder model are given in this section.

Datasets: The dataset description is given as follows:

UNSW-NB15 dataset [11]: Labeling features of 47 features are characterized in each record. The real-world traffic network packets have 47 features and network access of labelling features as normal or abnormal. The five groups are categorized in 47 features: additional generated features, time features, content features, basic features, and flow features.

CICIDS2017 [11]: The dataset of normal or attacks are captured from Monday, July 3, 2017, to Friday, July 7, 2017 of total of 5 days. The 80 network flow features of each records extracted from CICFlowMeter tool of generated network traffic.

NSL-KDD [11]: NSL-KDD dataset is derived from the KDD 99 dataset by eliminating redundant and duplicated records, which is more reasonable in both data size and data structure. The 42 of features, 118191 number of attacks, 67343 number of normal data instances, and 125973 number of sample data.

Kyoto 2006+ dataset [11]: The 24 number of features, 118191 number of attacks, 113120 number of normal data instances, 231311 number of sample sizes.

Parameter settings: In autoencoder parameter settings are adam optimizer is used, 64 batch size, 20 epochs, 3 hidden layers, and relu activation function were used. In SVM, the GO method selects C as 90.1, and 0.08 α for classification.

System information: Intel i9 processor system with 128 GB RAM, 22 GB GPU and Windows 10 64-bit OS. The GO-SVM and existing methods were evaluated on the same dataset and same environment.

Table 1. Quantitative analysis of AE-SVM-GO model

Methods	Accuracy (%)	DR (%)	FAR (%)
SVM	82.3	80.7	12.3
AE-SVM	86.3	87.1	8.7
SVM-GO	90.1	91.5	8.4
AE-SVM-GO	98.3	98.1	2.71

5. Results

The autoencoder model is applied to generate instances of minority classes to balance the dataset. The grasshopper optimization is applied for hyper-parameter optimization of SVM to performance attack classification in the network. The accuracy, detection rate (DR), and False alarm rate (FAR) were used to evaluate the AE-SVM-GO technique.

The quantitative analysis of the AE-SVM-GO model on intrusion detection is given in Table 1 and Fig. 2. The AE-SVM-GO model has the advantage of effectively handling the imbalance dataset and also provide optimal parameter settings for classification. The AE-SVM-GO model has higher DR and accuracy in intrusion detection systems. The SVM suffers from imbalance dataset and the AE-SVM model overcomes the imbalance problem. Still, parameter learning of the model is poor that affects the performance. The SVM-GO method increases the parameter learning in the SVM model and suffers from imbalance data problems. The AE-SVM-GO model has an accuracy of 98.3 %, the SVM-GO model has 90.1 %, AE-SVM has 86.3 %, and SVM has 82.3 % accuracy.

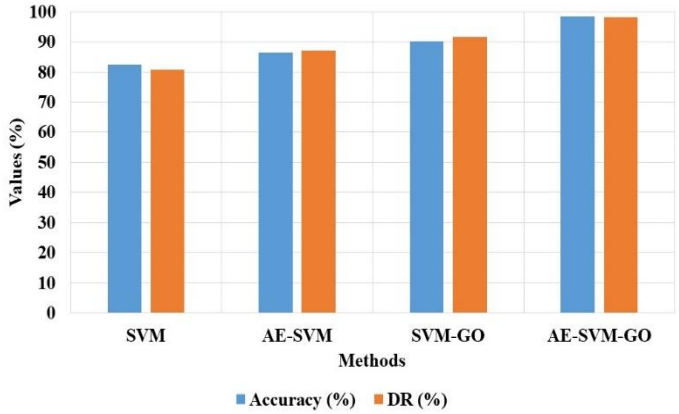

Figure. 2 AE-SVM-GO quantitative analysis on intrusion detection

Table 2. SVM hyper-parameter optimization in classification

Methods	Accuracy (%)	DR (%)	FAR (%)
SVM	86.3	87.1	8.7
SVM-GA	90.4	90.7	7.5
SVM-PSO	91.5	92.1	6.5
SVM-GWO	93.1	96.2	5.8
SVM-FF	96.2	95.4	4.2
AE-SVM-GO	98.3	98.1	2.71

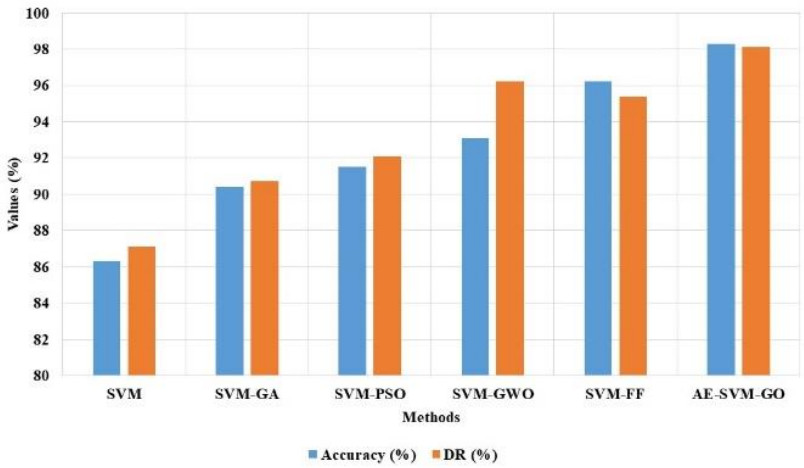

Figure 3. Hyper-parameter optimization of SVM in intrusion detection

Optimization methods such as grasshopper, particle swarm optimization (PSO), Firefly (FF), grey wolf optimization (GWO), and genetic algorithm (GA) were used for parameter optimization of AE-SVM, as given in Table 2 and Fig. 3. The Grasshopper model provides good convergence and escapes from local optima that improve the efficiency. The existing methods have limitations of easily being trapped into local optima and lower convergence in hyper-parameter optimization. The AE-SVM-GO has 98.3 % accuracy, FF has 96.2 %, GWO has 93.1 %, PSO has 91.5 %, GA has 90.4 %, and SVM has 86.3 % accuracy.

Standard classifiers such as random forest, long short term memory (LSTM), K-Nearest neighbors (KNN), Bi-LSTM and SVM were tested with grasshopper hyper-parameter optimization in intrusion detection, as given in Table 3 and Fig. 4. The SVM classifiers have higher efficiency due to improvement in learning performance and efficiency in handling many features. LSTM and RF models are suffers from the overfitting problem and hyper-parameter optimization improves its performance. The grasshopper model improves the KNN model and the KNN model has a limitation of outlier sensitivity. The RF has 92.4 %, KNN has 93.2 %, LSTM has 97.3 %, Bi-LSTM has 98.2 % and SVM has 98.3 % accuracy with hyper-parameter optimization in intrusion detection.

Table 3. Classifiers comparison with hyper-parameter optimization

Methods	Accuracy (%)	DR (%)	FAR (%)
RF-GO	92.4	92.5	5.1
KNN-GO	93.2	93.4	4.5
LSTM-GO	97.3	97.8	3.5
BiLSTM-GO	98.2	98.1	3.1
SVM-GO	98.3	98.1	2.71

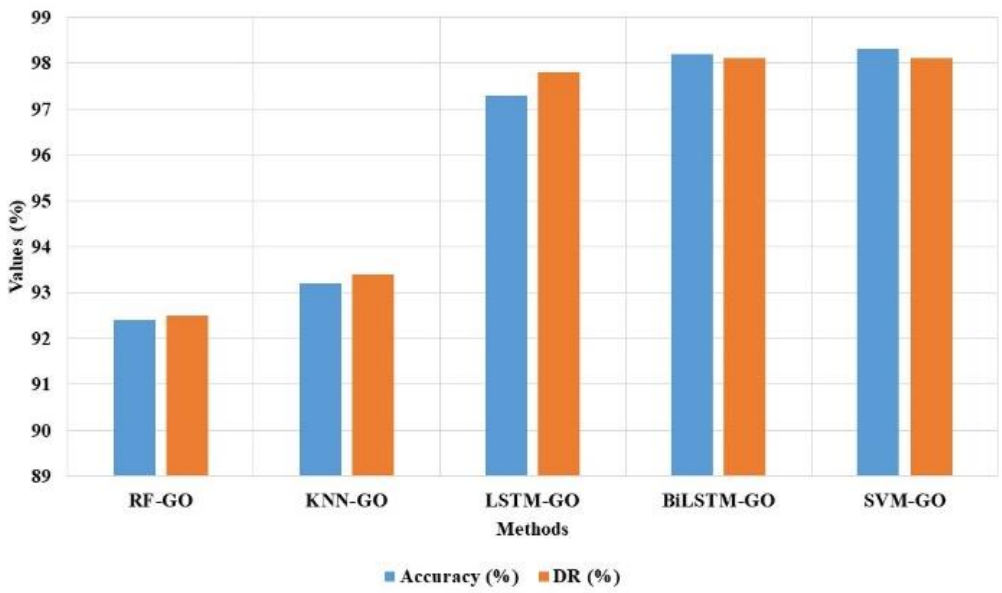

Figure 4. Classifier comparison for SVM-GO in intrusion detection

Table 4. Comparative analysis on various intrusion detection dataset

Methods	Datasets	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)	Training Time (s)
SVM-NB [11]	UNSW-NB15	93.75	92.4	91.5	91.95	64
	CICIDS2017	98.92	97.5	95.3	96.39	52
	NSL-KDD	99.35	98.2	97.1	97.65	48
	Kyoto 2006+ dataset	98.58	96.5	96.2	96.35	41
CNN-BiLSTM [12]	NSL-KDD	83.58	82.4	81.5	81.95	45
	UNSW-NB15	76.82	75.3	74.1	74.70	63
CRNN [13]	CSE-CIC-DS2018	97.6	96.2	96.1	96.15	51
AE-Triplet Network [14]	UNSW-NB15	92.4	91.4	91.2	91.30	62
ANN [15]	UNSW-NB15	91.2	90.4	90.2	90.30	60
AE-SVM-GO	UNSW-NB15	95.3	94.1	95.2	94.65	41
	CICIDS2017	99.2	99.1	99.1	99.10	36
	NSL-KDD	99.6	99.5	99.4	99.45	27
	Kyoto 2006+ dataset	99.1	99.1	99.1	99.10	16

5.1 Comparative analysis

The AE-SVM-GO model is compared with SVM-Naïve Bayes (NB) [11], CNN-BiLSTM [12], and CRNN [13] models in intrusion detection.

The AE-SVM-GO model is compared with existing research in intrusion detection in various datasets, as given in Table 4 and Fig. 5. The AE-SVM-GO model provides higher efficiency in intrusion detection due to the balance of dataset and learning of features with adaptive hyper-parameter in SVM. The CNN-BiLSTM [12] model suffers from an overfitting problem and SVM-NB [11] suffer from an imbalance data problem.

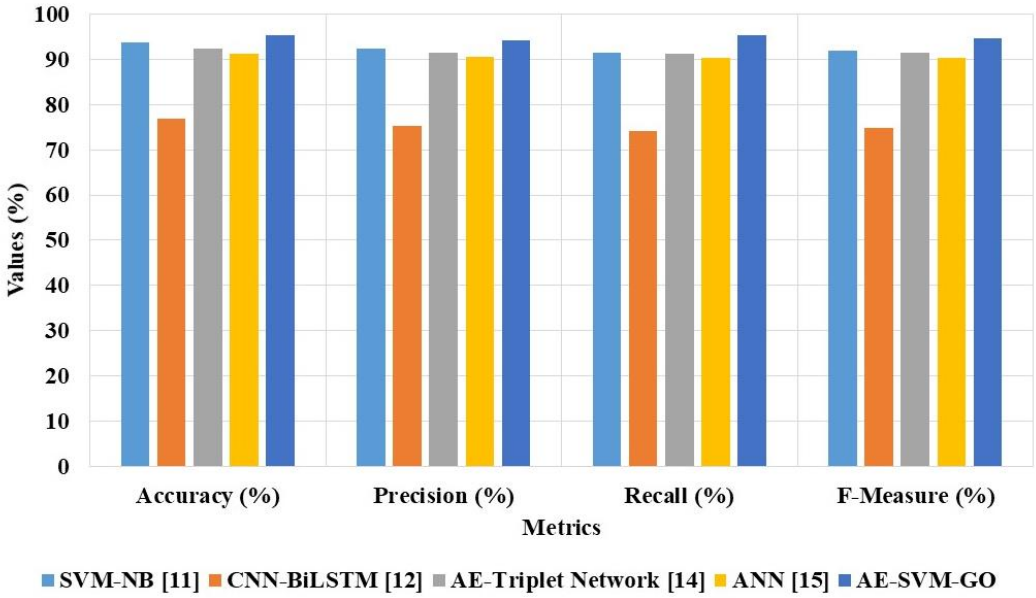

Figure 5. Existing method comparison on various intrusion detection dataset

6. Conclusion

Network intrusion detection system improves security and identifies the attacks in the network to eliminate the suspicious nodes in the network. The existing intrusion detection model has limitations of imbalance data problem. The autoencoder is applied in the AE-SVM-GO model to generate instances of minority classes in the dataset. The balanced dataset is applied to SVM with grasshopper method for parameter optimization to identify the intrusion in the network. The autoencoder model provides higher performance in balancing the dataset and improves the classification performance. Grasshopper method has higher performance than FF, GWO, PSO and GA in hyper-parameter optimization. The SVM model has higher performance in classification than RF, KNN, LSTM and BiLSTM due to its capacity to handle many features. The AE-SVM-GO method has accuracy of 95.3 %, recall of 95.2 %, and existing CNN-BiLSTM has accuracy of 76.82 %, and 74.1 % recall. Future work of this model involves applying IoT based network to identify the attacks and eliminate the suspicious nodes.

Notation List

Symbol	Description
A_i	wind advection
b	Bias
C	cost value
c	comfort area
$cmax$	maximum value
$cmin$	minimum value
d_{ij}	distance of two grasshoppers
$\hat{e}_g$	unity vector
$\hat{e}_w$	wind direction
f	Activation function
f	attraction intensity
G	Component
g	gravitational constant
G_i	grasshopper gravity force
H	Optimal hyper plane
$J(X,Z)$	Autoencoder objective train
l	attractive length scale
L	total iteration
l	current iteration
M	Total number of data instances
S	Training and Label data
s	social forces strength
S_i	social interaction
$\hat{T}_d$	target value
u	constant drift
W	Weight
x	input value
x_i	Input vector
x_max	maximum value
x_min	minimum value
x_N	normalized data
y	Output
y_i	Target value
z	Decoding output
α	population size

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

The paper background work, conceptualization, methodology, dataset collection, implementation, result analysis and comparison, preparing and editing draft, visualization have been done by first. The supervision, review of work and project administration, have been done by second author.

References

[1] Y. Pan, F. He, and H. Yu, "Learning social representations with deep autoencoder for recommender system", *World Wide Web*, Vol. 23, No. 4, pp. 2259-2279, 2020.

[2] R. V. Mendonça, A. A. Teodoro, R. L. Rosa, M. Saadi, D. C. Melgarejo, P. H. Nardelli, and D. Z. Rodríguez, "Intrusion detection system based on fast hierarchical deep convolutional neural network", *IEEE Access*, Vol. 9, pp. 61024-61034, 2021.

[3] M. Safaldin, M. Otair, and L. Abualigah, "Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks", *Journal of Ambient Intelligence and Humanized Computing*, Vol. 12, No. 2, pp. 1559-1576, 2021.

[4] Y. Imrana, Y. Xiang, L. Ali, and Z. A. Rauf, "A bidirectional LSTM deep learning approach for intrusion detection", *Expert Systems with Applications*, Vol. 185, p. 115524, 2021.

[5] H. Zhang, J. L. Li, X. M. Liu, and C. Dong, "Multi-dimensional feature fusion and stacking ensemble mechanism for network intrusion detection", *Future Generation Computer Systems*, Vol. 122, pp. 130-143, 2021.

[6] F. Folino, G. Folino, M. Guarascio, F. S. Pisani, and L. Pontieri, "On learning effective ensembles of deep neural networks for intrusion detection", *Information Fusion*, Vol. 72, pp. 48-69, 2021.

[7] G. Andresini, A. Appice, and D. Malerba, "Nearest cluster-based intrusion detection through convolutional neural networks", *Knowledge-Based Systems*, Vol. 216, p. 106798, 2021.

[8] X. Kan, Y. Fan, Z. Fang, L. Cao, N. N. Xiong, D. Yang, and X. Li, "A novel IoT network intrusion detection approach based on adaptive particle swarm optimization convolutional neural network", *Information Sciences*, Vol. 568, pp. 147-162, 2021.

[9] M. Mulyanto, M. Faisal, S. W. Prakosa, and J. S. Leu, "Effectiveness of focal loss for minority classification in network intrusion detection systems", *Symmetry*, Vol. 13, No. 1, p. 4, 2021.

[10] Z. Wang, Y. Zeng, Y. Liu, and D. Li, "Deep belief network integrating improved kernel-based extreme learning machine for network intrusion detection", *IEEE Access*, Vol. 9, pp. 16062-16091, 2021.

[11] J. Gu, and S. Lu, "An effective intrusion detection approach using SVM with naïve Bayes feature embedding", *Computers & Security*, Vol. 103, p. 102158, 2021.

[12] K. Jiang, W. Wang, A. Wang, and H. Wu, "Network intrusion detection combined hybrid sampling with deep hierarchical network", *IEEE Access*, Vol. 8, pp. 32464-32476, 2020.

[13] M. A. Khan, "HCRNNIDS: hybrid convolutional recurrent neural network-based network intrusion detection system", *Processes*, Vol. 9, No. 5, p. 834, 2021.

[14] G. Andresini, A. Appice, and D. Malerba, "Autoencoder-based deep metric learning for network intrusion detection", *Information Sciences*, Vol. 569, pp. 706-727, 2021.

[15] M. Choraś, and M. Pawlicki, "Intrusion detection approach based on optimised artificial neural network", *Neurocomputing*, Vol. 452, pp. 705-715, 2021.

[16] M. Yu, T. Quan, Q. Peng, X. Yu, and L. Liu, "A model-based collaborate filtering algorithm based on stacked AutoEncoder", *Neural Computing and Applications*, Vol. 34, No. 4, pp. 2503-2511, 2022.

[17] S. Zavrak, and M. İskefiyeli, "Anomaly-based intrusion detection from network flow features using variational autoencoder", *IEEE Access*, Vol. 8, pp. 108346-108358.

[18] J. Xu, and K. Duraisamy, "Multi-level convolutional autoencoder networks for parametric prediction of spatio-temporal dynamics", *Computer Methods in Applied Mechanics and Engineering*, Vol. 372, p. 113379, 2020.

[19] P. Tao, Z. Sun, and Z. Sun, "An improved intrusion detection algorithm based on GA and SVM", *IEEE Access*, Vol. 6, pp. 13624-13631, 2018.

[20] D. A. Otchere, T. O. A. Ganat, R. Gholami, and S. Ridha, "Application of supervised machine learning paradigms in the prediction of petroleum reservoir properties: Comparative analysis of ANN and SVM models", *Journal of Petroleum Science and Engineering*, Vol. 200, p. 108182, 2021.