Computer Systems Science & Engineering
DOI:10.32604/csse.2022.021851
Article

Tech Science Press

An Efficient Stabbing Based Intrusion Detection Framework for Sensor Networks

A. Arivazhagi[1,*] and S. Raja Kumar[2]

[1]Department of Computer Science and Engineering, University College of Engineering, Ariyalur, Tamilnadu, India
[2]Department of Mathematics, University College of Engineering, Ariyalur, Tamilnadu, India
*Corresponding Author: A. Arivazhagi. Email: arivu.ucea@gmail.com
Received: 17 July 2021; Accepted: 06 September 2021

Abstract: Intelligent Intrusion Detection System (IIDS) for networks provide a resourceful solution to network security than conventional intrusion defence mechanisms like a firewall. The efficiency of IIDS highly relies on the algorithm performance. The enhancements towards these methods are utilized to enhance the classification accuracy and diminish the testing and training time of these algorithms. Here, a novel and intelligent learning approach are known as the stabbing of intrusion with learning framework (SILF), is proposed to learn the attack features and reduce the dimensionality. It also reduces the testing and training time effectively and enhances Linear Support Vector Machine (*l*-SVM). It constructs an auto-encoder method, an efficient learning approach for feature construction unsupervised manner. Here, the inclusive certified signature (ICS) is added to the encoder and decoder to preserve the sensitive data without being harmed by the attackers. By training the samples in the preliminary stage, the selected features are provided into the classifier (*l*SVM) to enhance the prediction ability for intrusion and classification accuracy. Thus, the model efficiency is learned linearly. The multi-classification is examined and compared with various classifier approaches like conventional SVM, Random Forest (RF), Recurrent Neural Network (RNN), STL-IDS and game theory. The outcomes show that the proposed *l*-SVM has triggered the prediction rate by effectual testing and training and proves that the model is more efficient than the traditional approaches in terms of performance metrics like accuracy, precision, recall, F-measure, p-value, MCC and so on. The proposed SILF enhances network intrusion detection and offers a novel research methodology for intrusion detection. Here, the simulation is done with a MATLAB environment where the proposed model shows a better trade-off compared to prevailing approaches.

Keywords: Network security; sensor network; intrusion detection; learning framework; linear support vector machine; the detection mechanism

1 Introduction

Over the widespread of the Internet, there is extensive growth of devices connected with the broader range of networking technologies and communication that shows drastic change over the human lives [1].

These rapid technological growths are identified worldwide with the organizational functions like email systems, industrial applications, banking, and online shopping. However, these technological advancements have enhanced individuals' growth and change the world's perspective where information security seems to be crucial issues [2]. The industries need to offer a secured communication channel towards the internet users, including employees, customers, and identifying unauthorized functionalities. At present, Intrusion Detection Systems (IDS) provides a superior solution to security issues when compared to conventional network defence mechanisms like a firewall [3]. It assists the network administrator in detecting the attacks, breaches, vulnerabilities over the industrial network. There are two diverse forms of IDS. They are anomaly-based and signature-based IDS. In the former model, the system needs to categorize network traffic's unusual or unknown characteristics by examining the typical factors of network traffic structure [4]. There are some deviations from the usual traffic pattern to the network traffic and classified as an intrusion. The benefits of an anomaly-based model are the prediction of some newer and unknown attacks. Thus, the research concentrates on these types of intrusion detection system [5]. Similarly, the latter model predicts the attacks based on the rules pre-installed for attack identification over IDS. The network traffic is evaluated with the updated database of attack signatures for identifying intrusion in the available network traffic dataset [6].

These anomaly prediction approaches are utilized in diverse fields like fraud detection, network security, medical applications, and military applications. The following process is carried out to model efficient anomaly-based IDS [7]. Initially, some essential features need to be extracted, and dimensionality reduction has to be performed while extracting the subset of correlated features from the network traffic dataset to improve the classification outcomes [8]. Subsequently, some practical approaches are adopted for improving the classification outcomes and enhance the classification speed. There are variously supervised and unsupervised learning approaches to improve intrusion detection performance and improve prediction accuracy. ML approaches like support vector machine (SVM), self-organizing maps (SOM), random forest (RF), and decision tree (DT), which is adapted to categorize and identify an intrusion. Various investigators have investigated IDS using unsupervised learning approaches and shallow learning approaches like SVM, RF, and NB. These unsupervised learning approaches provide an enhancement in detection rate. Dimensionality reduction and unsupervised learning approaches are concurrently utilized for feature representation and feature extraction. It enhances data quality and improves the classification outcomes of conventional supervised learning approaches [9]. Currently, some remarkable achievements are identified over unsupervised deep-learning-based approaches using vision-based computing. Additionally, deep learning approaches are considered unsupervised learning approaches that assist supervised ML in enhancing performance and network traffic anomaly identification by diminishing the training and testing times [10]. Deep learning approaches have gained potential advancements to attain effectual data representation for constructing the enhanced models. Thus, with the adoption of deep learning techniques, the model is inspired by combining the cryptography-based security model. Here, Linear Support Vector Machine (*l*-SVM) is constructed with an auto-encoder method for feature construction in an unsupervised manner. An inclusive certified signature (ICS) is added to the encoder and decoder to preserve the sensitive data without being harmed by the attackers. By training the samples in the preliminary stage, the selected features are provided into the classifier (*l*-SVM) to enhance the prediction ability for intrusion and classification accuracy. This model intends to re-construct the input representation and transforms the feature representation of data associated with the input data. Thus, the performance is improved with considerable classification task. The significant contributions of this research work are listed as:

1) The experimentation begins with the acquisition of data from the online resources, i.e., NSL-KDD dataset. Then, the attack features are analyzed using the linear SVM model, which encodes features and separates the features with the hyperplanes. The feature dimensionality is reduced for eliminating

the computational overhead. It also helps in improving the prediction outcomes compared to conventional learning approaches like SVM and so on.

2) This work integrates the learning and cryptography concept to provide stability over the anticipated approach. Here, Linear Support Vector Machine (l-SVM) is constructed with an auto-encoder method for feature construction in an unsupervised manner. An inclusive certified signature (ICS) is then invoked with the encoder and decoder to preserve the sensitive data. In the preliminary stage, the chosen features are provided given to ($l-$SVM) for enhancing the prediction rate during intrusion detection and classification accuracy.
3) The efficiency of the anticipated model is analyzed with other approaches where the simulation is done with MATLAB 2020 environment. The predicted model shows a better trade-off when compared to different methods. It helps in the multi-class classification process. The training and testing are performed with the incoming network data, and the features are observed crucially for examining the process.

The remainder of the work is provided as follows: In Section 2, an extensive analysis is done with various prevailing approaches. The pros and cons related to those models are analyzed for providing an improved methodology. In Section 3, the anticipated model with its system flow is shown more extensively. Section 4 discusses the numerical values attained with the experimentation and the discussions related to it. Section 5 depicts the conclusion of the provided model with the limitations identified during the process of computation. The ideas for future research are also discussed to motivate the young researchers to make a better achievement in research.

2 Related Works

The modernization of security measures is often considered to be constrained based on the resource nature of WSN. However, resources are exhausted and lead to the duplication of security approaches are observed and fail to protect the secured sensors over the sensitive field. For this case, lightweight security measures with minor resource consumption are appropriate SNs. Thus, there should be a proper balance between the cost and security with added countermeasures. Based on the attacker's nature, various countermeasures are anticipated and dedicated to the security prevention models. Moreover, these approaches are not provided with higher-capacity influencing nature and therefore need to merge it with the security requirements [11]. Piotrowski et al. [12] discuss key establishment using the asymmetric cryptographic model, and it provides stable and secure key generation mechanism, key distribution and authentication. Subsequently, WSN does not fulfil the asymmetric cryptographic requirements with memory storage and computing capacity. A diverse investigator anticipates the probability of adopting public-key cryptography with the reduction of computational complexity, stored data, and transmitted data. The author considers Elliptic Curve cryptography as the finest example for these sorts of researches. Ian et al. [13] discuss various other symmetric cryptographic approaches used for securing WSN: MD5, SHA-1, IDEA, RC4, and RC5. The cryptographic based symmetric key provides more significant features for low-cost energy and speed. The most appropriate key management approaches rely on the asymmetric cryptographic model. Often, this symmetric key cryptography provides key exchange issues. The general solution is adopting pre-distribution models where the keys are loaded over the SNs for deployment.

Zhu et al. [14] anticipate a key management protocol model for SNs with some essential principles. The expected protocol offers excellent compromise among the scalability and robustness. Thus, the robustness of the model is improved by incrementing the number of keys. The author does not provide an obvious explanation regarding the revoking process and updating keys. It does not assist cluster-based models and does not fulfil the nodes with standard access and the neighbours. Therefore, the nodes mustn't be accessible. The numbers of compromised nodes are incremented, and the securities offered by these approaches are not so appropriate. Panja et al. [15] anticipates the improved version of the public key concept. The investigator adopts the matrix key with the simple symmetric key idea to select pair-wise simple key concepts. This model is highly robust against compromised attacks and offers a potential energy consumption degree. Zhang et al. [16] anticipate authentication protocol and localized encryption model. It relies on a hybrid key distribution model with the utilization of four diverse kinds of key establishment: cluster-key, group key, key* pair, and a single key. Some keys are unique, and it is utilized for establishing secured communication among the base station. Yuan et al. [17] discuss SHELL protocol to secure the member nodes and CH with the benefit of robustness towards the compromise nodes. The capturing nodes are accountable for network scalability and nodes replacement. Gandino et al. [18] discuss hierarchical key management protocol for key grouping. The preliminary idea is to generate a secret key from the partial key concept. It is the probable key establishment, modifying, and revoking the partial keys. This model is simple, more accessible for execution, reduces computational complexity, storage space, and energy consumption. Thus, it is vulnerable to compromising the attacks. The secured hierarchical key management concept is discussed by [19] to fulfil the WSN requirement. It is hugely more straightforward, ease of communication and storage and provides higher-level security. Fakhrey et al. [20] discuss the equilibrium model for predicting the optimal distribution for maximizing pre-distribution key concept, lifetime, connectivity, resilience, and cost. It is scalable, and it is easier for analyzing various applications based key-management model. Sun et al. [21] discuss dynamic key management and dynamic mechanism over heterogeneous WSN. The essential concept behind the lightweight protocol model is based on key establishment and authentication to provide a higher level of security. Omar et al. [22] discuss the novel multiple location-based key management protocols for random distribution. It helps to overcome the consequences of the compromised entity. The local-dynamical scheme is anticipated along with layered-cluster topology to attain key management [23–26]. It saves higher energy with energy reduction for protecting the network security. The ultimate goal is to reduce the number of nodes for key updation periodically [27].

3 Methodology

Here, an extensive analysis is performed to provide security and intends to overcome the unnecessary feature patterns that show significant consequences over the network model. A linear Support Vector Machine is constructed with an auto-encoder model with efficient learning approaches for determining the features in an unsupervised manner. A cryptosystem known as an inclusive certified signature is used to preserve sensitive data from malicious users. The performance is evaluated using various other conventional approaches, and simulation is done with MATLAB 2020 environment. For validation purpose, the NSL-KDD dataset is used for testing and training the samples. Fig. 1 depicts the framework of the SILF model.

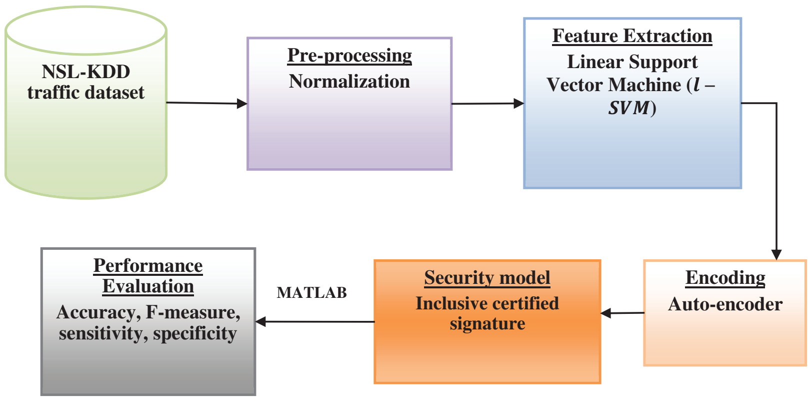

Figure 1: Block of FTTS-CNN model

3.1 NSL-KDD

Tavallaee et al. [28] recommend NSL-KDD dataset in 2009, which various inherent disadvantages identified over the KDD-CUP'99 dataset. The comparison of various protocol models are shown in Tab. 1 and the categories of the attacks are shown in Tab. 2.

Table 1: Comparison of various protocol models [29]

Protocol	Simplicity	Scalability	Robustness	Energy consumption	Objective
LEAP	Moderate	Moderate	Moderate	Moderate	Secure routing
SHELL	Low	Moderate	High	Moderate/High	Trust link establishment
SEHKM	Moderate	Moderate	Moderate	Moderate/Low	Malicious node isolation
EDAK	High	Moderate	High	Low	Secured data sharing
LAMP	Low	Moderate	Moderate	High	Attack prediction
Local dynamic scheme	Low	High	Moderate	Moderate	Authentication
Key management scheme	Moderate	High	Moderate	Moderate	Data filtering
Symmetric bi-variate polynomials	Low	Moderate	High	Moderate	Clustered security
Basis	Moderate/ low	Moderate	High	Moderate/Low	Secured access control
Asymmetric key generation	Low	Low	High	Moderate	Secured data routing

Table 2: Attack types and categories

Type	Categories
U2R	Ps, Xterm, Perl, Sqlattack, rootkit, load module, buffer_overflow
R2L	Named, send_fmail, httptunnel, snmp_getattack, snmp_guess, xsnoop, xlock, spy, waremaster, ware-relient, imap, ftp_write, phf, multi-hop, guess_password
Probe	Saint, mscan, portsweeop, ipsweep, satan
DoS	Worm, udpsworm, process_table, smurf, tear_drop, apache_2, land_neptune, back

The above-given dataset holds various training and testing set records. The total number of records for training includes 1, 27,973 and testing records contains 22,544. The traffic record consists of 41 features with 35 continuous and six symbolic features. Tab. 2 depicts the feature categories like basic, content and traffic, and Tab. 3 describes the classified dataset like U2R, R2L, probing and denial of service attacks. The difference among the testing and training dataset shows a realistic basis for detecting intrusions.

3.2 Pre-processing

Here, dataset normalization is performed over the non-numeric features like flag, service, and protocol-type. The non-numeric features are converted to numeric features with diverse attributes like ICMP, UDP, and TCP, i.e., (0, 0, 1), (0, 1, 0), (1, 0, 0) over binary-vectors. The service feature includes 70 attributes, and flag features include 11 attributes. Normalization is depicted as the difference between the maximum and minimum values. The feature mapping is normalized with max-min normalization and mathematically expressed as in Eq. (1):

$$x_i = \frac{x_i - min}{\max - min} \tag{1}$$

From Eq. (1), x_i specifies data point, *max* specifies maximal values from data points for all features, and *min* specifies minimal values of all data points.

Table 3: NSL-KDD feature description

Features	S. No	Names	Data_type	Features	S. No	Names	Data_type
Basic	1	Duration	Continuous	Content	21	is_host_login	Symbolic
	2	Protocol_tye	Symbolic		22	Is_guest_login	Symbolic
	3	Service	Symbolic	Traffic	23	Count	Continuous
	4	Flag	Symbolic		24	server_count	Continuous
	5	Source_bytes	Continuous		25	serror_rate	Continuous
	6	Destination_bytes	Continuous		26	server_serror_rate	Continuous
	7	Land	Continuous		27	rerror_rate	Continuous
	8	Wrong_fragment	Continuous		28	server_rerror_rate	Continuous
	9	Urgent	Continuous		29	same_reeror_rate	Continuous
	10	Hot	Continuous		30	different_server_rate	Continuous
Content	11	Number_failed_logins	Continuous		31	server_different_host_rate	Continuous
	12	Logged_in	Symbolic		32	destination_host_count	Continuous
	13	Number_compromised	Continuous		33	destination_host_server_count	Continuous
	14	Root_shell	Continuous		34	destination_host_same_server_rate	Continuous
	15	Su_attempted	Continuous		35	destination_host_different_server_rate	Continuous
	16	Number_root	Continuous		36	destination_host_same_source_port_rate	Continuous
	17	Number_file_creations	Continuous		37	destination_host_server_different_host_rate	Continuous
	18	Number_shells	Continuous		38	destination_host_serror_rate	Continuous
	19	Number_access_files	Continuous		39	destination_host_server_serror_rate	Continuous
	20	Number_outbounded_commands	Continuous		40	destination_host_rerror_rate	Continuous
					41	destination_host_server_rerror_rate	

3.3 Auto-Encoder

The preliminary architecture of an auto-encoder with several dimensions over the input and output layer; however, the hidden layer possesses fewer dimensions (see Fig. 2). The auto-encoder model is composed of input data with learned information over a compressed manner. The model provides better flexibility over neural network models, and the auto-encoder is modelled with several hidden layers variation and number of nodes [30–34]. The encoder model is provided with an input of $x_i \in R^{d_x}$ and reduces the $y_i \in R^{d_y}$ over the hidden layer with the adoption of function $f(.)$ using identity function for linear projection $f(x) = \frac{1}{1+e^{-Wx}}$ for non-linear mapping where $'W'$ is $d_y * d_x$ based weighted matrix. The bias of the NN is avoided, and the encoding process is mathematically expressed as in Eq. (2):

$$y_i = f(Wux_i) \tag{2}$$

Here, W^T is $d_y * d_x$ based weighted-matrix, and decoding is performed with the below-given expression as in Eq. (3):

$$x_i' = g(W^T y_i) \tag{3}$$

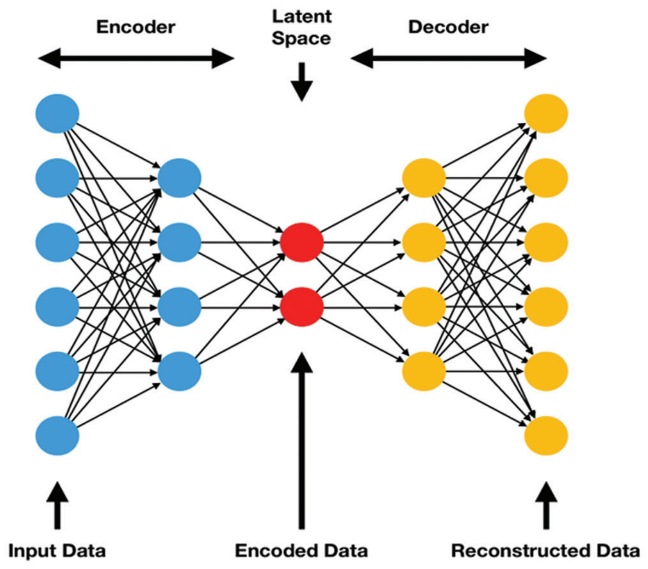

Figure 2: Auto-encoder model

From Eq. (3), $g(\cdot)$ is sigmoid function with non-linear reconstruction or reconstruction of linear function *f*(.). It is utilized for specifying the decoding process, and d_x is the input dimension, and d_y is the output dimension after reducing dimensionality. Here, R^{d_y} specifies output dimensions of output vectors, and R^{d_x} is the set of input data vectors dimension. The re-construction of the decoder is provided with the set of instances and indexed by Ω with weights of $S_i = s_{ij}, s_{ik}$ for x_i with weighted re-construction error as in Eq. (4):

$$e_i = \sum_{j \in \Omega_i} s_{ij} L(x_j . x_j') \tag{4}$$

The weighted re-construction error for all the input samples towards the auto-encoder is expressed as in Eq. (5):

$$E = \sum_{i=1}^{i=n} e_i(W, \quad W^T) \tag{5}$$

The auto-encoder iteratively computes, and S_i value is updated using the linear SVM model, which is discussed below. The objective of utilizing an auto-encoder is to reduce the total weighted re-construction error minimally. It is performed iteratively till it reaches the convergence point. Here, auto-encoders are used for extracting (see Fig. 3). The original form of features and to improve the classification accuracy. The reconstructed data is fed as an input to the classifier to enhance the prediction accuracy.

3.4 Construction of Linear Support Vector Machine ($\mathbf{l - SVM}$)

Generally, SVM has handled both binary classification and regression problems. SVM performance is validated successfully in various applications. SVM pretends to find the optimal hyperplane that categorizes the data points by separating the points of two different classes, as in Fig. 4. In linear-separable form, SVM separates hyperplane with a larger margin. It is expressed as in Eqs. (6) and (7):

$$x_i^T w + b \geq 1; \quad for\ all\ y_i = +1 \tag{6}$$

$$x_i^T w + b \leq \ -1; \ for\ all\ y_i = -1 \tag{7}$$

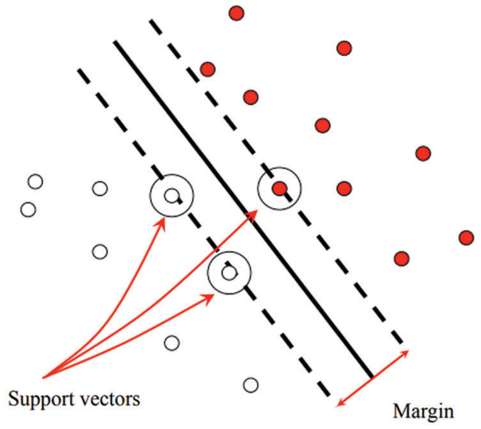

Figure 3: Support vectors and margins for feature extraction

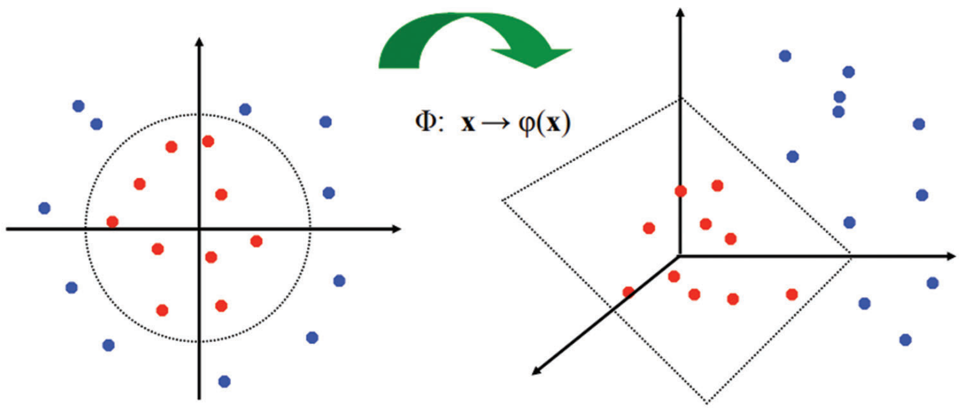

Figure 4: Mapping high dimensional data

The hyperplane intends to separate the data iff it satisfies the given condition in Eq. (8):

$$y_i(x_i^T w + b) \geq 1; \quad \forall i \tag{8}$$

Here, 'X is input pattern; 'w' is a weighted vector; 'b' is bias. The weight and bias are evaluated with a maximal margin of $1/\|w\|$, subjected to training patterns and outside margin. It is expressed as in Eqs. (9) and (10):

$$\min \frac{1}{2} ||w||^2 \tag{9}$$

$$y_i(x_i^T w_b) \geq 1; \quad i = 1,\ 2,\ \ldots,\ N \tag{10}$$

Here, $y_i \in \{-1, 1\}$ is training pattern labels. The data needs to be separated with a maximal marginal hyperplane. $\|w\|^2$ is minimized to $w^T w$. It is expressed as in Eq. (11):

$$Min_{(w,b)} \left\{ \frac{||w||^2}{2} \right\}; \quad y_i(w^T x_i + b) \geq 1; \quad i = 1,\ 2,\ 3,\ \ldots,\ n \tag{11}$$

Here, (x_i, y_i) is training samples in training data, and 'n' is several training instances. With Lagrangian duality, it is expressed as in Eq. (12):

$$Max\ F(\lambda_i) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2}\ ||w||^2 \tag{12}$$

$$= \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j x_i^T x_j \tag{13}$$

Here, $\sum_i^{\lambda} y_i = 0;\ \ \lambda_i \geq 0;$ for $i = 1, 2, \ldots, n$. λ_i is Lagrange multiplier. The optimal hyperplane decision function is expressed as in Eq. (14):

$$f(x) = sign \sum_{j=1}^{n} \lambda_i y_i (x_i^T x) + b \tag{14}$$

Similarly, in non-separable cases, the data cannot be separated correctly. A slight modification is carried out to partition training data with minimal error. In minimization problem, slight variations are provided with positive slack variable $\xi_i \geq 0$. It is expressed as in Eq. (15):

$$Min_{(w,b,\xi)} \left\{ \frac{1}{2}\ ||w||^2 + C \sum_{i=1}^{n} \xi_i \right\};\ \ y_i(w^T x_i + b) \geq 1 - \xi_i \tag{15}$$

Here, $i = 1, 2, \ldots, n$; $'C'$ is regularization parameter. The feature selection process needs to be reduced as huge features leads to higher time consumption and computational complexity.

3.5 *Inclusive Certified Signature for the Security Model*

Here, an inclusive certified signature model is explained to establish security to the incoming data over the network model. It includes security parameter 1^k as input and returns the certifier's private key and system parameters over the intrusion detection system. It preserves the private keys while publicly accessible incoming traffic data in the SILF. When the user intends to maintain the incoming data, the system takes the input from the system parameters and identity ID_s. It shows the output with secret key values S_{IDS} and public key PK_{IDS}. The inclusive certified signature model takes the input system with a specific technique, private key, and identity IDs of users and public key PK_{IDS}, and it gives the user's certificate information, inclusive certificate, and components of the user's inclusive certificates. It runs the model with inclusive certification, which is shared secretly among the users'. Similarly, some system parameters like user's certificate information, private key, and secret keys are associated with full public keys for exclusive certified signature. The output is shown as exclusive certificate $cert_{IDS}$ and sets to the user through the public network channel.

The private key setting uses some system parameters like user's certificate information CI_{IDS}, secret key S_{IDS}, public key PK_{IDS}, implicit certificate $\overline{SK}_{IDS},$ and other components. Then, it returns the user's private key $SK_{IDS} = (S_{IDS},\ \overline{SK}_{IDS})$. The signer is executed to generate the signature $'\sigma'$ by considering the input, message, and user's certificate information with key pairs (SK_{IDS}, PK_{IDS}), and corresponding components of inclusive and exclusive certificates. The signature verification is performed to validate the signature with the input message, signature pair, CI_{IDS}, Pk_{IDS}, R_{IDS}, A_{IDS}, m, *and* σ, $cert_{IDS}$. The outcome is to validate the message signature. Else, the process is not proper.

3.6 *Stabbing of Intrusion with Learning Framework (SILF)*

Suppose public/private key pair's Sk_{IDS}, Pk_{IDS} are provided to the legitimate users. In that case, it is unfeasible for any adversary to evaluate the valid signature of the user without knowing the private key. Diverse kinds of adversaries are considered as they intend to spoil the security framework of the network model. During the attack scenario, the adversaries are allowed to get the signer to sign any incoming messages and fail to generate the valid message signature pair. When the adversary is considered an unauthorized user who pretends to compromise the target node with secret key S_{IDS} or eliminates the public key PK_{IDS}. but, it fails to get access to the inclusive digital certificate or private key attached over the incoming data. In some cases, the adversaries need to provide a security model over the non-certified users or malicious users, i.e., the users without certificate or registration over the certificate model. Another security scenario is where the malicious users can access the master key while it cannot replace the public keys. The assumption is considered to be accountable where the adversaries with the master key can impersonate anyone. It is more specific to view the certified users, i.e., users who have possession over the exclusive digital certificate $Cert_{IDS}$ and key-pairs (Pk_{IDS}, Sk_{IDS}), before the master key becomes a known factor for the adversaries. The term stabbing is framed to avoid or to eradicate the occurrence of attacks over the network. When the occurrence of attacks are reduced, then the quality of network functionality is increased to fulfil the end-users need.

4 Evaluation Metrics

This work adopts the NSL-KDD dataset for validating the functionality of SILF for enhancing intrusion detection over the network. The experimentation is done on the Intel (R) core i5 processor at 2.71 GHz with 8GB RAM. The simulation is done in MATLAB 2020, where the kernel is considered as the classifier model. Various performance metrics are adopted to evaluate SILF performance, and the attributes outcomes from the training and testing model are utilized in assessing the performance measure. It is defined as True Positive (TP) - sample anomaly classified appropriately as an anomaly; True Negative (TN): standard samples classified appropriately as normal; False Positive (FP): normal sample classified wrongly as an anomaly; and False Negative (FN): sample anomaly instances classified wrongly as normal. The performance metrics are computed as follows:

1) Accuracy- specifies the appropriate proportion of total records in the NSL-KDD testing set. It is mathematically expressed as in Eq. (16):

$$\textbf{Accuracy} = \frac{\textbf{TP} + \textbf{TN}}{\textbf{TP} + \textbf{TN} + \textbf{FP} + \textbf{FN}} \tag{16}$$

2) Precision- specifies the proportion of properly predicted intrusion to the predicted intrusion over the testing process. It is mathematically expressed as in Eq. (17):

$$\textbf{Precision} = \frac{\textbf{TP}}{\textbf{TP} + \textbf{FP}} \tag{17}$$

3) Recall: specifies the proportion of properly predicted intrusion to the total intrusion samples over the testing set. It is mathematically expressed as in Eq. (18):

$$\textbf{Recall} = \frac{\textbf{TP}}{\textbf{TP} + \textbf{FN}} \tag{18}$$

4) F-measure: It is the measure of both recall and precision as shown in Eq. (19):

$$\textbf{F}-\textbf{measure} = \frac{2*\textbf{Precision} * \textbf{Recall}}{\textbf{Precision} + \textbf{Recall}} \tag{19}$$

The experimentation is performed to examine the effectiveness and efficiency of the lower-level features provided to the $\boldsymbol{l-SVM}$ classifier model with binary class ($'0' - \boldsymbol{normal}$, $'1' - \boldsymbol{anomaly}$) multi-classes ($\boldsymbol{probe}$ $\boldsymbol{U2R}$, $\boldsymbol{R2L}$, $\boldsymbol{and}$)regular over the provided dataset. Moreover, training and testing are performed to compute the efficiency of the anticipated SILF model. It intends to address intrusion detection with the cryptosystem, reduce computational complexity, and fulfil the storage requirements. The extraction of the most acceptable data representation and lower-level feature dimensionality is the model's primary focus. The number of support vectors needs to be reduced as the kernel model requires computational space and memory linearly, directly proportional to support vectors. The SILF model performance is compared with general approaches like SVM, Random Forest (RF), Naive Bayesian (NB), J48, and Decision Tree. Testing and training are done separately for evaluating the significance of the model with low-dimensionality features. With the theoretical analysis of the SILF model, it is observed that the hidden number and the parameters are highly influenced by classification accuracy and training speed. The optimization of hyperparameters is more challenging for modelling an efficient deep learning model for detecting intrusion. Investigations of the hidden units with sparse parameters are more challenging and reduce testing and training time. The performance of $\boldsymbol{l-SVM}$ is increased with the search for the best parameters. The tuning of hyper-parameters is done with auto-encoders on the training NSL-KDD dataset. The hyper-parameter values are chosen, trained, and tested using the anticipated SILF as it shows better performance over the binary classification process when $\boldsymbol{\beta}=3$, $\boldsymbol{\lambda}=0.000001$, $\boldsymbol{p}=0.50$. The models also get higher prediction accuracy over multi-class classification with $\boldsymbol{\beta}=3$, $\boldsymbol{\lambda}=0.000005$, $\boldsymbol{p}=0.78$.

Tab. 4 depicts the accuracy anticipated SILF with prevailing models like game theory, single SVM, STL-IDS, RF, and RNN. The experimentation of the anticipated SILF model is done with 5 to 30 iterations. The accuracy of SILF over 30th iteration is 95.78% which is 25.58%, 19.019%, 15.3%, 15.11% and 12.5% respectively (See Fig. 5). Tab. 5 depicts the comparison of precision where the SILF shows 100%, which is 7.02% and 6.08% higher than single SVM and STL-IDS, respectively (See Fig. 6). Tab. 6 depicts recall where the SILF model shows 89%, 27.66% and 20.72% higher than SVM and STL-IDS, respectively. Tab. 7 depicts the comparison of F-measure where the SILF model shows 94.17%, which is 17.77%, 19.89%, and 15.1% higher than game theory, single SVM, and STL-IDS, respectively (See Fig. 7). Tab. 8 shows the time comparison of the proposed SILF model with other approaches like single SVM and STL-IDS, respectively. The execution time is 1325.424006 s, 3503.118 s, and 1362.384 s (See Fig. 8). Tab. 9 depicts the comparison of detection rate of 0.0042 s (See Fig. 10). Fig. 11 depicts the Min objective *vs*. The number of function evaluations where the minimal observed objective and the estimated minimal objectives are computed. During the experimentation process, the encoder function is investigated to analyze dimensionality reduction and feature learning to improve prediction accuracy and reduce $\boldsymbol{l-SVM}$ training and testing time. The prediction accuracy is higher when the hidden units are 30 with 1000 epochs. At last, to validate the significance of the encoder-parameter, training speed and classification is required. The validation of the training data using $\boldsymbol{l-SVM}$ accuracy is 9.621110e+01, recall is 9.166667e+01, and precision is 9.977221e+01 with an error rate of 3.788904e-02. The validation of testing data using $\boldsymbol{l-SVM}$ gives an accuracy of 9.578544e+01, recall of 89, precision of 100, and an error rate of 4.214559e-02. Fig. 9 shows the kernel size of the anticipated model. The elapsed time of the anticipated SILF model is 1325.424006 s. The above analysis shows that the anticipated model works in predicting the intrusion over the network model using learning and cryptosystem models.

Table 4: Accuracy comparison

Iterations	5	10	15	20	30
SILF	81.88	86.97	90.78	92.57	95.78
Game theory	53	59.11	62.3	68.5	70.2
Single SVM	80.32	76.74	84.96	81.4	76.761
STL-IDS	65.24	69.65	71.23	76.87	80.48
Random forest	68.97	72.48	76.87	78.25	80.67
RNN	70.14	76.58	79.58	81.45	83.28

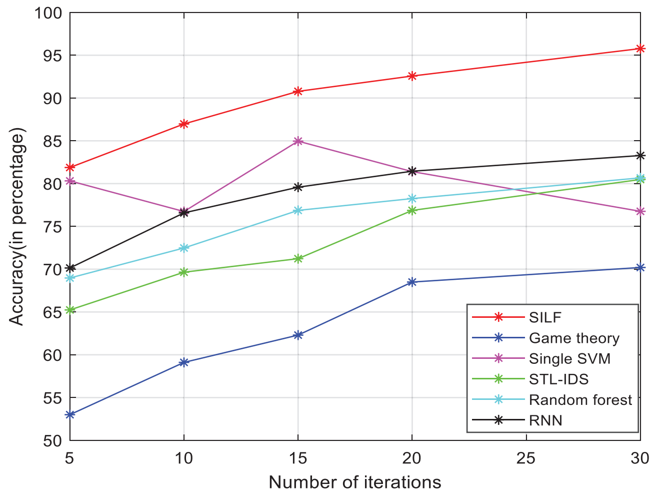

Figure 5: Accuracy comparison

Table 5: Precision comparison

Iterations	5	10	15	20	30
SILF	75.24	86.97	92.56	95.66	100
Single SVM	71.80	78.12	86.55	90.47	92.98
STL-IDS	76.8	80.47	87.41	92.58	93.92

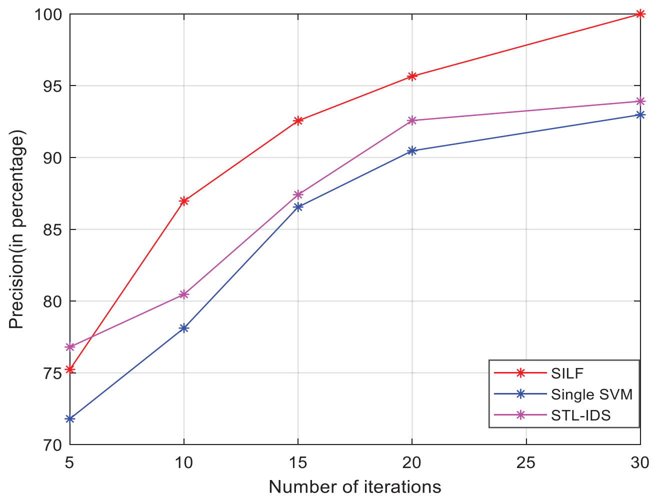

Figure 6: Precision comparison

Table 6: Recall comparison

Iterations	5	10	15	20	30
SILF	65.98	76.11	81.52	86.47	89
Single SVM	50.78	55.68	58.12	60.25	61.84
STL-IDS	51.02	56.77	60.85	63.99	68.28

Table 7: F-measure comparison

Iterations	5	10	15	20	30
SILF	72.58	79.56	82.98	89.92	94.17
Game theory	52.22	57.9	61.55	65.89	76.4
Single SVM	55.58	60.04	62.89	69.87	74.28
STL-IDS	59.22	63.22	67.58	72.45	79.07

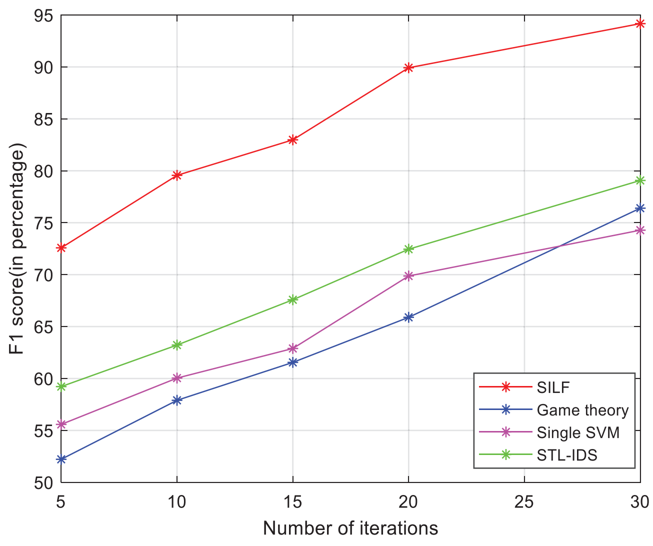

Figure 7: F1-score comparison

Table 8: Time comparison

Iterations	5	10	15	20	30
SILF	305.22	400.98	529.47	1125.14	1325.424006
Single SVM	709.455	1362.84	1867.87	2601.873	3503.118
STL-IDS	413.85	465.992	673.031	1611.34	1362.384

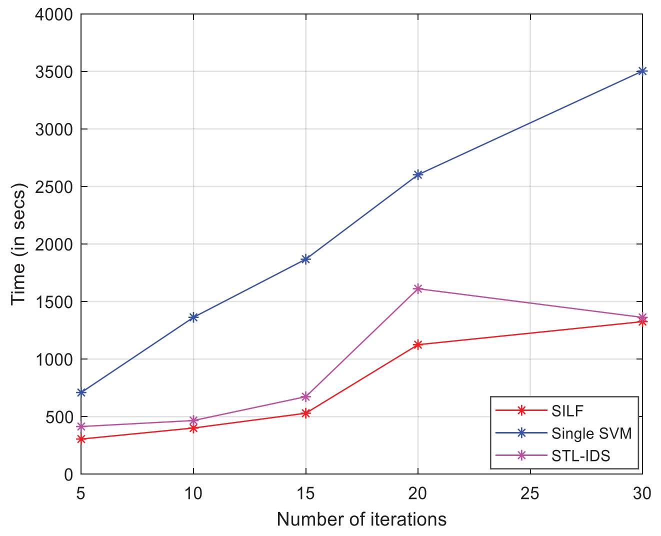

Figure 8: Time in seconds evaluation

Table 9: Detection rate evaluation

Iterations	5	10	15	20	30
Missing detection rate	0.007	0.0062	0.0059	0.0048	0.0042

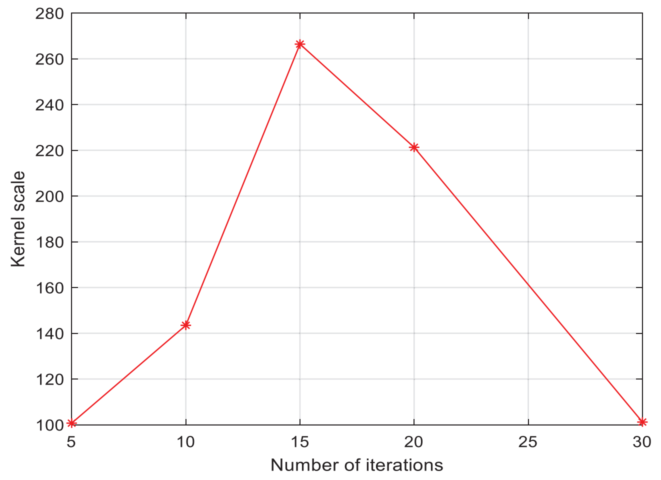

Figure 9: Kernel size computation

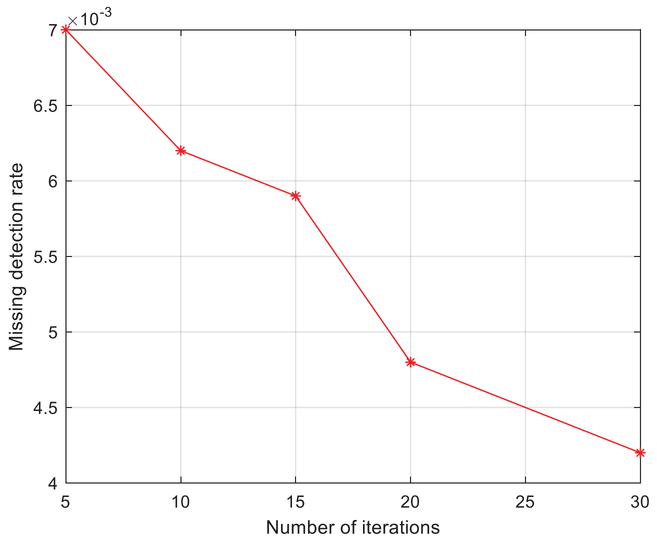

Figure 10: Detection rate

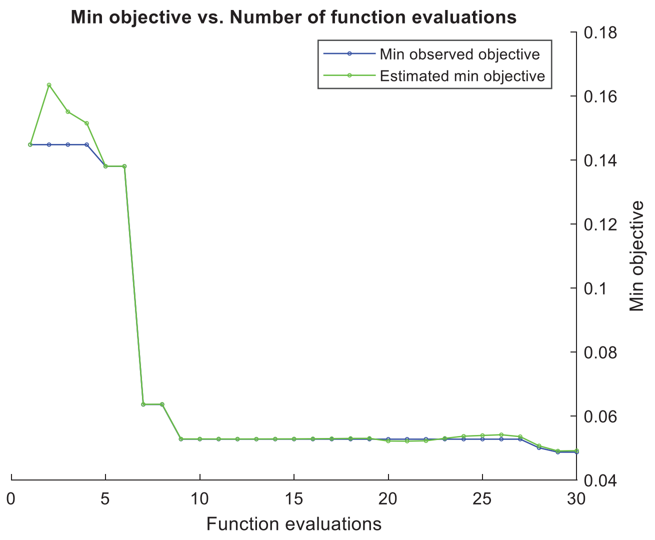

Figure 11: Min objective *vs*. number of function evaluations

5 Conclusion

The anticipated SILF uses linear SVM and auto-encoder for efficient learning and feature construction using an unsupervised manner. For enhancing security, and inclusive certified signature is used and provides protection over sensitive data. Thus, the incoming data is not harmed by malicious users. In the preliminary stage, the samples are trained and select appropriate features to improve the prediction rate for enhancing classification accuracy. The anticipated framework provides better results compared to conventional approaches that lag in analyzing the linearity of the model, only concentrating on the non-linear measures. The extensive analysis is done with the use of the NSL-KDD dataset over a multi-class classification process. The evaluation is done with the available benchmark dataset to reduce the time complexity and cost-efficiency. The SILF model objective is achieved with better security and helps to overcome the vulnerabilities. Thus, SILF can be adopted over any sort of real-time applications to enhance security. In future, the anticipated model is provided with the construction of a multi-stage pipelined SILF framework by hybridized feature analysis for superior feature representation and dimensionality reduction model. The training and testing time is reduced with the pipelined execution over the parallel computing GPU platform.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] N. Pereira and R. M. de Moraes, "Comparative analysis of AODV route recovery mechanisms in wireless Ad hoc networks," *IEEE Latin America Transactions*, vol. 8, no. 4, pp. 385–393, 2010.

[2] S. Seo, S. Park and J. Kim, "Improvement of network intrusion detection accuracy by using restricted boltzmann machine," in *Proc. of 8th Int. Conf. on Computational Intelligence and Communication Networks (CICN)*, Tehri, India, 2016, pp. 413–417.

[3] K. Haseeb, N. Islam, A. Almgren and I. U. Din, "Intrusion prevention framework for secure routing in WSN-based mobile internet of things," *IEEE Access*, vol. 7, pp. 185496–185505, 2019.

[4] F. N. Nur, S. Sharmin, M. A. Habib, M. A. Razzaque and M. S. Islam, "Collaborative neighbour discovery in directional wireless sensor networks," in *Proc. of IEEE Region Conf. (TENCON)*, Cheju, South Korea, 2017, pp. 1097–1100.

[5] F. N. Nur, S. Sharmin, M. A. Habib, M. A. Razzaque, M. S. Islam *et al.,* "Collaborative neighbor discovery in directional wireless sensor networks: Algorithm and analysis," *EURASIP Journal on Wireless Communications and Networking*, vol. 1, pp. 1–15, 2017.

[6] J. Yan, M. Zhou and Z. Ding, "Recent advances in energy-efficient routing protocols for wireless sensor networks: A review," *IEEE Access*, vol. 4, pp. 5673–5686, 2016.

[7] V. V. Mandhare, V. R. School and R. R. Manthalkar, "QoS routing enhancement using metaheuristic approach in a mobile ad-hoc network," *Computer Networks*, vol. 110, pp. 180–191, 2016.

[8] H. Shi, Y. Zhang, Z. Zhang, N. Ma, X. Zhao *et al.,* "Hypergraph-induced convolutional networks for visual classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 10, pp. 2963–2972, 2019.

[9] P. Sherubha, S. P. Sasirekha, V. Manikandan, K. Gowsic and N. Mohanasundaram, "Graph based event measurement for analyzing distributed anomalies in sensor networks," *Springer Sådhanå*, vol. 1, no. 45, pp. 1–5, 2020.

[10] P. Sherubha and N. Mohanasundaram, "An efficient network threat detection and classification method using ANP-MVPS algorithm in wireless sensor networks," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 11, pp. 1597–1606, 2019.

[11] P. Sherubha and N. Mohanasundaram, "An efficient intrusion detection and authentication mechanism for detecting clone attack in wireless sensor networks," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 11, no. 5, pp. 55–68, 2019.

[12] K. Piotrowski, P. Langendoerfer and S. Peter, "How public key cryptography incense wireless sensor node lifetime," in *Proc. of Fourth ACM Workshop on Security of ad hoc and Sensor Networks*, New York, United States, 2006, pp. 169–176.

[13] F. B. Ian, G. Seroussi and N. P. Smart, "Advances in elliptic curve cryptography," *London Mathematical Society Lecture Note Series*, London, United Kingdom, 2005. pp. 1–228.

[14] S. Zhu, S. Setia and S. Jajodia, "LEAP + Efficient security mechanisms for large-scale distributed sensor networks," *ACM Transactions on Sensor Networks*, vol. 2, no. 4, pp. 500–528, 2006.

[15] B. Panja, S. K. Madria and B. Bhargava, "Energy and communication efficient group key management protocol for hierarchical sensor networks," in *Proc of. IEEE Int. Conf. on Sensor Networks, Ubiquitous, and Trustworthy Computing*, Taichung, Taiwan, 2006, pp. 384–393.

[16] X. Zhang and J. Wang, "An efficient key management scheme in hierarchical wireless sensor networks," in *Proc. of Int. Conf. on Computing, Communication and Security*, Pointe aux Piments, Mauritius, 2015, pp. 1–7.

[17] Q. Yuan, C. Ma, X. Zhong, G. Du and J. Yao, “Optimization of key pre-distribution protocol based on supernetworks theory in heterogeneous WSN,” *Tsinghua Science and Technology*, vol. 21, no. 3, pp. 333–343, 2016.

[18] F. Gandino, R. Ferrero, and M. Rebaudengo, “A key distribution scheme for mobile wireless sensor networks: q–s–Composite,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 34–47, 2017.

[19] S. Athmani, A. Bilami and D. E. Boubiche, “EDAK: An efficient dynamic authentication and key management mechanism for heterogeneous WSNs,” *Future Generation Computer Systems*, vol. 92, pp. 789–799, 2019.

[20] H. Fakhrey, M. Johnston, F. Angelini and R. Tiwari, “The optimum design of location-dependent key management protocol for a multiple sink WSN using a randomly selected cell reporter,” *IEEE Sensors Journal*, vol. 18, no. 24, pp. 10163–10173, 2018.

[21] B. Sun, Q. Li and B. Tian, “Local dynamic key management scheme based on layer-cluster topology in WSN,” *Wireless Personal Communications*, vol. 103, no. 1, pp. 699–714, 2018.

[22] M. Omar, I. Belalouache, S. Amrane and B. Abbache, “Efficient and energy-aware key management framework for dynamic sensor networks,” *Computers & Electrical Engineering*, vol. 72, pp. 990–1005, 2018.

[23] Y. Liu and Y. Wu, “A key pre-distribution scheme based on sub-regions for multi-hop wireless sensor networks,” *Wireless Personal Communications*, vol. 109, no. 2, pp. 1161–1180, 2019.

[24] S. I. Chu, Y. J. Huang and W. C. Lin, “Authentication protocol design and low-cost key encryption function implementation for wireless sensor networks,” *IEEE Systems Journal*, vol. 11, no. 4, pp. 2718–2725, 2015.

[25] K. A. Shim, “Basis: A practical multi-user broadcast authentication scheme in wireless sensor networks,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 7, pp. 1545–1554, 2017.

[26] R. Amin, S. K. H. Islam, G. P. Biswas and M. S. Obaidat, “A robust mutual authentication protocol for WSN with multiple base stations,” *Ad Hoc Networks*, vol. 75, pp. 1–18, 2018.

[27] D. Wang and P. Wang, “Two birds with one stone: Two-factor authentication with security beyond conventional bound,” *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 708–722, 2018.

[28] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in *IEEE Symp. on Computational Intelligence for Security and Defence Applications*, Ottawa, ON, Canada, 2019, pp. 1–6.

[29] D. Wang, W. Li and P. Wang, “Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4081–4092, 2018.

[30] F. Farahnakian and J. Heikkinen, “A deep auto-encoder based approach for intrusion detection system,” in *Proc. of 20th Int. Conf. on Advanced Communication Technology*, Chuncheon, South Korea, 2018, pp. 178–183.

[31] P. Muneeshwari and M. Kishanthini, “A new framework for anomaly detection in NSL-KDD dataset using hybridneuro-weighted genetic algorithm,” *Journal of Computational Science and Intelligent Technologies*, vol. 1, no. 1, pp. 29–36, 2020.

[32] C. Narmatha, “A new neural network-based intrusion detection system for detecting malicious nodes in WSNs,” *Journal of Computational Science and Intelligent Technologies*, vol. 1, no. 3, pp. 1–8, 2020.

[33] S. L. Sarah and L. A. Mahmoud, “A novel intrusion detection system in WSN using hybrid neuro-fuzzy filter with ant colony algorithm,” *Journal of Computational Science and Intelligent Technologies*, vol. 1, no. 1, pp. 1–8, 2020.

[34] A. N. Suresh, “A hybrid genetic-neuro algorithm for cloud intrusion detection system,” *Journal of Computational Science and Intelligent Technologies*, vol. 1, no. 2, pp. 15–25, 2020.