ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 1, 2024

Evolution Deep Belief Network for Cyber – Attack Detection in Industrial Automation and Control System

Dr. ADELINE JOHNSANA, Associate Professor, Department Of CS SICET, Hyderabad
G SRIKANTH, PASHAPU MITHIL, DASARLA MADHU YADAV, CIRASA NAGANDLA BHARADWAZ
UG Student, Department Of CS, SICET, Hyderabad

Abstract

Supervisory control and data acquisition (SCADA) stands as a control system consisting of computers and networked data communications. At present, many industries use SCADA to monitor as well as control the processes. In recent days, numerous attacks are targeting these systems. Thus, the furtherance of high-security SCADA is much-needed one on account of its susceptibility to attacks centered on the architectural restriction. To identify these attacks, numerous classifications, optimization methods, and intrusion detecting systems (IDS) are posited. The chief drawbacks of this prevailing work are detecting accuracy, high training time, and security. For prevailing over these disadvantages, an NK-RNN classifier is proposed to recognize the intrusions in the SCADA method. Initially, the features from the datasets are organized, and the important attributes are chosen by utilizing the Elephant Herding Optimization (EHO). Secondly, the data, which is optimized, are grouped and classified by applying the NK-RNN classifier. Then, the outcomes, which are classified, are assessed and utilized to outcome prediction. In normal data, Caesar Ciphering is employed for the prevention of attacks and also the modified elliptic curve cryptography is employed for enhancing the security level. From the performance assessment, it is revealed that the NK-RNN method attains superior performance than the prevailing classification method along with IDS algorithms.

Keywords Supervisory control and data acquisition · Attack detection · Intrusion detection system · Elephant Herding Optimization algorithm · Feature selection · Normalized K-means clustering algorithm

1 Introduction

A SCADA is basically an Industrial Controls System (ICS) that functions in public and also private industrial processes counting significant infrastructures. These systems depend more and more on information and communications technology (McEvoy and Wolthusen 2011; Lin et al. 2017). In several countries, networked and also completely gated irrigations were furnished with SCADA to facilitate communications, sensing, and also controlling (Amin et al. 2012) industrial processes, like oil mining, water treatment plants, electric grids, traffic control systems, space stations in addition to nuclear systems (Ghosh and Sampalli 2019). Usually, a SCADA will have a control server that is deployed at the control center, one or several topographi- cally distributed field sites encompassing field devices, along with communication links. In addition, it will be made of Programmable Logics Controllers (PLCs) (Cher- dantseva et al. 2016), Remotes Telemetry Units (RTUs) (Samdarshi et al. 2015), Human Machines Interface (HMI) (Yılmaz and Gönen 2018), which render user interaction with the operator to monitor and also control the complete system, along with these, it also comprises a collection of networked devices, say sensors, controllers, actuators, along with communication devices (Li et al. 2016). SCADA protocols that are implemented in big geographi- cal regions have Ethernet/IP, Modbus, Profinet, DNP3, DCOM, et cetera. Communication is done by means of satellite, cellular networks, radio or microwaves, switched telephone or lease-line communication media by these protocols on a Wide Areas Network (WAN) (Upadhyay and Sampalli 2020).

Although there are numerous advantages with Internet, like scalability, lucrative, better communications protocols, effectiveness, interoperability betwixt elements and remote access, attacks, say Denial of Service (DoS), Wormhole Attack, Probe, Users to Roots (U2R) as well as Remotes to Locals (R2L) (Hemdan and Manjaiah 2018) as of Internet can eliminate the great advantages of the SCADA. Network connectivity along with security was not considered in the design of SCADA (Nazir et al. 2017). Since SCADA communication has homed big connectivity with non-proprietary networks via the internet, the cyber-security problems are augmenting in it (Shahzad et al. 2015a, b). The chief challenge that the SCADA faces is the Detection and classification of intrusions in addition to attacks (Shitharth et al. 2020). The efficacy of SCADA system information security relies on the implemented protection technologies of transport environment data transmission elements (Finogeev and Finogeev 2017). IDS are software or tool to actively detect the attacker. The attack behavior along with the intrusion information is amassed to attain better fortification of the sensor network (Abusafat et al. 2018). Initially, the SCADA using a sensor amasses the data as of the distributed processes and considered it as the input. The system will hoard this big data subsequent to pre-processing (Enescu and Bizon 2017). Thus, the inputted dataset is pre-processed. This phase consists of repeat data removal, Replacement of missing attributes, as well as normalization (Shitharth 2017). In feature selection (FS), the superfluous features are eradicated utilizing feature extraction techniques, like Linear Weighted Cuckoo Search Optimization (LWCSO), Intrusion Weighted Particle based Cuckoo Search Optimization, enhancement of Mutual Information Features Selection (MIFS), and Modified Mutual Information-Centered Feature Selection (MMIFS) (Ambusaidi et al. 2016) et cetera. Choosing the optimum feature set will lessen the memory along with time consumption (Krishnan Sadhasivan and Balasubramanian 2017). Once the optimal features are chosen, this is taken into the classifier to categorize the data as an attack or normal data. And if it is a normal data, then prevention steps are set-about with the help of encryption and decryption (Hassan 2019). The author's contribution towards this work is summarized as:

1. For lessening the feature's dimensionality for better classification, a collection of classification and intrusion detection (ID) algorithms is used. Here, the hybridization of normalized K-Means clustering along with recurrent neural networks (RNN) is the NK-RNN classifier that is utilized. When analogized to the other existent methods, the detection effectiveness of attacks is enhanced.
2. Utilizing EHO, feature selection is performed for choosing significant attributes.
3. For the prevention of attacks, Caesar Ciphering is used and the modified elliptic curve cryptography (MECC) is used for improving the security level (SL).

This work is systematized as Sect. 2 offers the associated work. Section 3 proffers a concise discussion. Simulation outcomes are inferred in Sect. 4, and the chief findings of the paper are deduced in Sect. 5.

2 Literature survey

The literature (Goldenberg and Wool (2013)) recommended model-centered IDS on the basis of the key scrutiny for SCADA that looked deep into Modbus/TCP packets and generated an extremely meticulous traffic model. The method was extremely susceptible and was capable of flagging anomalies. The IDS was analyzed on a production Modbus. Notwithstanding its higher compassion, the system encompassed an extremely low false-positives rate. Additionally, the IDS effectively flagged anomalies, which were brought about by means of technicians who were revamping the HMI. The system also aided in identifying programmable logics controllers (PLC), which were configured wrongly. The literature Yang et al. (2014) developed SRID for detecting the intrusion within SCADA. The chief defense attention was mainly on the false data injections attack. The SRID detected these attacks and deduced the feasible attack origins in an effectual means. Additionally, a graph-centered detection model was commenced that joined the state alternation vectors along with the state relation graph. As of the assessment outcomes, the design effectively detected numerous data injections attacks and also inferred attack origins.

A multiple-layer cyber-security scheme for a future SCADA-particular IDS modelled by the literature Yang et al. (2014). The system analyzed manifold aspects for rendering an inclusive solution that alleviated different cyberattack threats. The multi-aspects IDS comprised a heterogeneous white list along with behavior-centered conception for making SCADA cyber systems safer. A multilayer cyber-security centered upon IDS was also recommended for protecting SCADA cybersecurity in smart grids devoid of encompassing the normal data availability. The chief pros of the framework were to guarantee power delivery as safe, stable, along with reliable. The literature Almalawi et al. (2015) suggested innovative clustering-centered IDS for detecting SCADA

customized attacks. This was centered upon a data-driven clustering of process parameters that automatically recog- nized the normal as well as decisive states of a specified system. After that, it extracted proximity-centered detec- tion rules as of the recognized states for monitoring rea- sons. This approach's effectiveness was analyzed by experimenting '8' datasets that comprised process param- eters' values. The empirical outcomes illustrated a 98% average accuracy in automatically recognizing the decisivestates whilst easing the monitoring of the SCADA.

The literature Shahzad et al. (2015a, b) looked at the security of SCADA systems along with protocols, partic- ularly the SCADA/DNP3 protocol. To meet the study's objectives, a SCADA simulation environment for water pumping was created using intelligent sensor connectivity, the payload was built, security was implemented inside the DNP3 protocol stack, and then bytes were broadcast to sub- controllers. The security performances were authenticated in opposition to attacks detection percentage and attacks impact percentage, which evaluated the considerable security enhancements. The literature Lin et al. (2018) presented a semantic analysis that incorporated the networkIDS with a power flow examination that could assess the implementation penalty of control commands. An adaptive power flow analysis was executed for balancing detection accuracy with the latency and precise detection of malev- olent control commands was perceived as of susceptible SCADA network. A 0.8% false-positive rate along with a 0.01% false-negatives rate was produced by the adaptive power flow analysis algorithm. The detection was com- pleted by the semantic analysis in approximately 200 ms, even in the instance of the large-scale test system.

Cyberattack detection centered on temporal patterns recognition presented by the literature Kalech (2019). These methods not merely looked for anomalies in the datapassed by the SCADA through the network but as well looked for anomalies that occurred by mis-utilizing gen- uine commands in order that illegal and erroneous time intervals betwixt them might cripple the system. Specially, '2' algorithms were proffered that was centered upon Hidden Markov Models as well as Artificial Neural Net- works (ANN) and estimated the algorithms on real as well as simulated SCADA data with '5' disparate feature extraction techniques; in each one, the algorithms regarded disparate facets of the data (raw). The outcomes illustrated that this method, particularly those centered upon time feature extraction, detected cyber-attacks, even those that concerned genuine functions that were recognized was tough to detect. The literature Shlomo et al. (2020) sug- gested '2' machine learning algorithms. A supervised algorithm was the first one that found common temporal patterns. Then, it was identified in the SCADA communi- cation protocols' data payload and employed as features in a classification method. The unsupervised algorithm was the second. It learned an automaton that represented the system's temporal behavior. Unknown states or events were then labeled harmful at runtime. The first supervised system, which used frequent temporal patterns as features, performed better than a baseline method that examined the mean along with standard deviation, according to an experimental evaluation using the genuine MODUBS- SCADA dataset as of Ben-Gurion University. The deficit of documentation concerning BGU SCADA networks restricted the capability of verifying the data and creating meaningful injections. The literature Lu et al. (2021) proffered a population extremal optimization (PEO)-cen- tered deep belief network detection approach (PEO-DBN) for detecting the cyber-attacks of SCADA-centered IACS. For identifying the DBN's parameters comprising the number of hidden units, the size of mini-batch, along with the learning rate, the PEO algorithm was utilized as there was no clear knowledge to choose these parameters. For enhancing the single method's performance for cyber-at- tack identification, the ensemble learning strategy for aggregation of the recommended PEO-DBN method named EnPEO-DBN was presented. By contrasting some prevailing methods, the proposed detection techniques were assessed on the gas pipeline system dataset along with the water storage tank system dataset as of SCADA net- work traffic. Via performance analysis, the superiority of PEO-DBN along with EnPEO-DBN was exhibited by the simulation results. The methods discussed several prob- lems like frequency changes in the system, computational cost, and security malfunctioning.

3 Proposed methodology

These days, in remote monitoring as well as physical processes controlling in contemporary Critical Infrastruc- tures, the ICS of SCADA is extensively employed. Fur- thermore, it collects the data concerning the physical process state as of a remote location and sends the equiv- alent commands for controlling the physical process. Of lately, these systems are being the center of augmenting attacks. This paper proposes efficient attack detection along with prevention system in SCADA. The chief objective of this paper is to get accurate Attack Detection (AD), process time reduction, and security enhancement. For this exact purpose, a Normalized K-Means clustering is hybridized with RNN, which is called NK-RNN classifier. The pro- posed NK-RNN based IDS include the following stages: preprocessing, FS, AD (classification), Ciphering, and Encryption. Initially, the network dataset is offered as an input for preprocessing. Subsequently, the significant fea- tures are chosen by using the EHO algorithm. Then, the

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 1, 2024

classification is performed, wherein the NK-RNN classifier classifies whether the data are attacked one or not. If it is a normal data, then the data undergoes ciphering and encryption for a security purpose. In a ciphering process, the original data is transmuted into cipher using Caesar cipher. Also, encryption and decryption are carried out using the MECC algorithm. The block diagram of the proposed work is exemplified in Fig. 1.

3.1 Dataset

The data taken as of the Intel Berkeley's research laboratory are gathered for the untreated measurements, which have both normal and attacked data. The sensors in this dataset are redistributed on indoor and also outdoor locations. Employing sensors to perceive the function of the network along with its protocols renders cognizance that eases more effective use of resources on account of the data amassed by the sensors. Each record encompasses '41' features, say protocol, class, source bytes, land, duration, flag, et cetera in this dataset. In the proposed work, IDS pre-process this dataset. This system is trained for the AD. The proposed NK-RNN based IDS detects the external attacks, say Probe (the miscreant will scan the networking system of any vulnerabilities or weaknesses to exploit it in the future for compromising the system. This commonly affects the data mining field, for instance, saint, mscan, portsweep, nmap, et cetera.), DoS (the miscreant endeavors to deplete user resources to a point where the user is impuissant to requite to other service requests. For instance, apache, Neptune, mail bomb, ping of death, UDP storm, smurf, back, etc.), U2R (the miscreant creates a user account as normal and then tries to exploit the system's vulnerabilities to attain the prerogative of super-user, for instance, xterm, perl.), along with R2L (a miscreant can send a packet to a machine on the network without encompassing any account in the network. The miscreant accesses the victim machine as a local user by utilizing the system's weaknesses).

3.2 Preprocessing

Data pre-processing stands as a vital step in AD. This will enhance the dataset's quality by transmuting it to a format that is simpler and efficiently processed for user comprehension. This also intends to lessen the data size, and find the relation betwixt data, normalize data and remove outliers. The preprocessing in the proposed work follows '3' phases: (i) remove repeated data, (ii) replace missing attributes and (iii) normalization, to simplify the dataset. These '3' steps are elucidated in the sections given below.

3.2.1 Remove repeated data

The data that were all same will bring about the same outcome, thus, the repeated data are removed as of the dataset in this phase. Together with that, the processing time will also be more, thus, by taking these repeated data off, the processing time will be reduced with increased speed.

3.2.2 Replace missing attributes

If some values are missing on the record, then those missing attribute should be replaced with the average value for that particular attribute.

3.2.3 Normalization

Normalization systematizes the data so as to have more effective access. By implementing the normalization, an efficient outcome can well be attained by the system. The

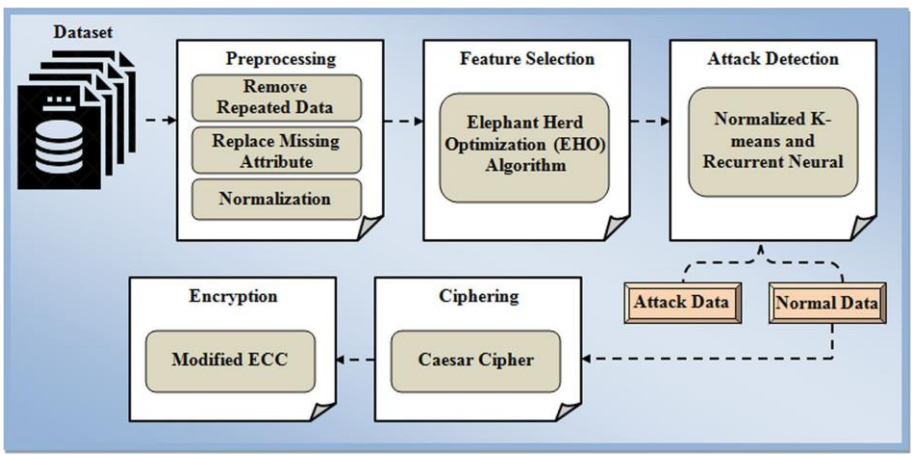

Fig. 1 Overall flow of the proposed NK-RNN based IDS

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 1, 2024

data values are attuned to a particular range, for instance, within [0, 1] or [− 1, 1] to obtain normalization. Thus, the proposed work adjusts the data value to the range [0, 1]. Since the limited dataset is utilized and the variability betwixt minimal and maximal is insignificant, the Min– Max normalization is considered here. The normalized value (D_{norm}) for every attribute data (D) is computed by means of the subsequent formula,

$$D_{norm} = \frac{\eth D - D_{min}\text{Þ}}{\eth D_{max} - D_{min}\text{Þ}} \quad \omega\ \eth 1 - 0\text{Þ}\ \text{þ}\ 0 \qquad \eth 1\text{Þ}$$

where, D_{min} is the minimum value of data, and D_{max} is the maximum value of data.

3.3 Feature selection

Numerous sensors are being included in the SCADA, which will augment the network's memory intricacy. Thus, the optimal selection of features is needed to trounce this problem. This will also ameliorate the AD's performance, which can be performed by choosing the pertinent features and eradicating redundant as well as irrelevant features from the dataset. Devoid of forfeiting the accuracy, FS simplifies a dataset by means of evading the curse of dimensionality, reducing overfitting, as well as identifying pertinent fundamental features. The execution speed might augment with the reduction of irrelevant features. Here, the EHO Algorithm is employed for FS. The meticulous elu- cidation of the EHO algorithm is rendered in Sect. 3.3.1.

3.3.1 Elephant Herding Optimization algorithm

EHO is basically a meta-heuristic swarm-centered search algorithm that solves an assortment of optimization issues. This algorithm mimics the herding behavior of elephants. With the help of the ensuing simplified rules, the EHO can be elucidated in a simple way.

1. Elephants from disparate clans cohabit and matriarchwill lead this group. There will be precisely the same number of elephants in each clan.
2. A fixed set of male elephants will depart from their family and live their life solitarily.
3. For the optimization issue, a matriarch will be theperfect elephant of the clan.

The EHO steps are elucidated as follows,

1. Generate individuals m and bifurcate the populace into n clans. Afterward, compute the fitness value for every individual and arrange the entire individuals as per their fitness.
2. Update every individual's position on the clan c_n. Let's presume that the clan is implied as c_n and the subsequent position of every solution m in clan c_n is updated as,

$$x_{new;c_n;m} = x_{c_n;m}\ \text{þ}\ a \times \eth x_{best;c_n} - x_{c_n;m}\text{Þ} \times c \qquad \eth 2\text{Þ}$$

where, $x_{new;c_n;m}$ is the newly updated position of the solution m in clan c_n, $x_{c_n;m}$ is the old position of the solution m in clan c_n, and $x_{best;c_n}$ is the position of the best solution in clan c_n. a represent the scale factor that ascertains the effect of $x_{best;c_n}$ on $x_{c_n;m}$. c 2 [0, 1] implies an arbitrary number as of the uniform distri- bution. Choose and retain the best solution betwixt $x_{new;c_n;m}$ and $x_{c_n;m}$ by employing Eq. (2).

3. Update $x_{c_n;m}$ and generate $x_{new;c_n;m}$ for attaining the finest solution by utilizing Eq. (3) if the solution m's position is equivalent to the best solution position ($x_{c_n;m} = x_{best;c_n}$). The fittest solution in every clan can well be updated as:

$$x_{new;c_n;m} = b \times x_{center;c_n} \qquad \eth 3\text{Þ}$$

where, b 2 [0, 1] implies a factor that ascertains the effect of the $x_{center;c_n}$ on $x_{new;c_n;m}$. Conversely, the new individual $x_{new;c_n;m}$ in Eq. (3) is produced by the information attained by the complete solutions from clan c_n. Choose and retain the best solution betwixt $x_{new;c_n;m}$ and $x_{best;m}$. $x_{center;c_n}$ signifies the center of clan c_n, and for the dth dimension, it can well be computed as,

$$x_{center;c_n;d} = \frac{1}{N_{c_n}} \times \sum_{m=1}^{N_{c_n}} x_{c_n;m;d} \qquad \eth 4\text{Þ}$$

where, N_{c_n} is the Number of solution in clan c_n.$1 \le d \le D$ signifies the dth dimension, in addition, D represents its total dimension.

4. Swap the worst fitness individual in clan c_n by utilizing separating operator as,

$$x_{worst;c_n} = x_L\ \text{þ}\ \eth x_U - x_L\ \text{þ}\ 1\text{Þ} \times r \qquad \eth 5\text{Þ}$$

where, $x_{worst;c_n;m}$ is worst individual in clan c_n, x_L is lower bounds of the individual position, and x_U is Upper bounds of the individual position. And r [[0, 1] implies a sort of stochastic distribution as well as uniform distribution in the gamut [0, 1].

5. By means of the newly updated positions, assess the populace and gauge the fitness for every solution. Return the best solution (s_f) amongst the entire clans asper their fitness value. The proposed EHO pseudo-codeis exhibited in Algorithm 1.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 1, 2024

Input: Dataset features
Generate individuals; Divide population into n clans; Calculate fitness of each individual; Set generation counter
Output: Optimized features
Initialize maximum generation MaxGen.
While t < MaxGen do
Begin Sort all the individuals according to their fitness.
Clan Update:
for all clan c_n **do**
for all solution m in the clan c_n **do**
Update $x_{c_n,m}$ and generate $x_{new,c_n,m}$
Select and retain best solution between $x_{c_n,m}$ and $x_{new,c_n,m}$
Update $x_{best,m}$ and generate $x_{new,c_n,m}$
Select and retain best solution between $x_{best,m}$ and $x_{new,c_n,m}$
end for
end for
for all clans c_n in the population do
Replace the worst solution in clan c_n
end for
Evaluate population and calculate fitness
end while
return best solution among all clans
End

Algorithm 1: Pseudo-code for the proposed EHO

After that, the chosen features are clustered as well as classified for detecting attacks that are briefly elucidated in the below segment.

3.4 Attack detection using NK-RNN algorithm

Subsequent to FS, the clustering along with classification processes are executed for detecting whether the data is normal or attacked data. The NK-RNN is posited for an effective AD. The chosen features are inputted to NK-RNN aimed at clustering as well as classification. The initial centroid point selection in the existing K means algorithm has more outliers. This outlier skews cluster grouping and increases the amount of time needed to find an optimal
solution. To solve this problem, the normalized weight value will be applied to the initialization of centroid in the existing k-means algorithm to avoid the detection of out- liers. The existing attack detection system trains non-se- quential data. For improving the attack detection system accuracy, the clustered data will be trained sequentially using the RNN algorithm. Initially, NK-RNN does clus- tering, and then those of objects in such a means that objects in the same group are more alike (in some sense) to one another than to those in other groups is termed as the clustering. Here, NKMA cluster the optimized data. A meticulous elucidation con- cerning the NKMA is provided in the section below.

3.4.1 Normalized K-means clustering algorithm

In this phase, NKMA clusters the optimized data centered upon the adjoining sensor's values. Initially, it takes the chosen features of the dataset that comprises the minimum along with maximum values. Thus, normalization is implemented to these data (with chosen features) that ameliorate the system's accuracy. The normalization is written as follows,

$$D_{sf_{norm}} = \frac{D_{sf} - D_{sf_{min}}}{D_{sf_{max}} - D_{sf_{min}}} \quad \text{ð6Þ}$$

clustered values are classified for detecting attacks. The procedure of grouping a compilation where, $D_{sf_{min}}$ is minimum value of data, and $D_{sf_{max}}$ is maximum value of data.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 1, 2024

After normalization, KMA initializes several clusters along with centroid. It is the uncomplicated un-supervised learning algorithms, which resolve the recognized cluster- ing issue. KMA intends to split the observations into clusters wherein every observation is an afflicted to the cluster with the adjoining mean, serving as an archetype of the cluster. The KMA's steps are elucidated as,

1. State the number of clusters K and after that initializes the K number of feature values as f¼ ff ;1f ;2. . .f g together with the set of cluster centroid as c_i ¼ fc1; c2; .. .cng.
2. Decide on the number of 'K' cluster centers arbitrarily to cluster the chosen features.
3. Gauge the distance betwixt each data point (DP) and the entire centroids, and allocate every point to the adjoining center whose distance from the cluster center is lesser of all the cluster centers.
4. Compute the clusters' centroid by means of averaging the complete DP that belongs to every cluster.
5. Repeat step 3 centered upon the new centroids. If the allotment of the cluster aimed at the DP's changes, then repeat step 3 otherwise, end the process.

The distance betwixt the DP is computed utilizing Euclidean distance (ED). The ED between '2' points x_1 ¼ ðx11; x12; .. .x1nÞ and x_2 ¼ ðx21; x22; .. .x2nÞ is assessed as,

$$E_d(x_1, x_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2} \quad (7)$$

Limit the square error of the distance between a data point x_{1i} and center c_i of its cluster as specified by the square error function offered by the Eq. (8).

$$s_e = \sum_{i=1}^{n} \sum_{j=1}^{n} \left\| E_d(x_{1i}, c_i) \right\| \quad (8)$$

Herein, s_e implies the square error function, $E_d(x_{1i}, c_i)$ is the ED between x_{1i} and c_i, and K implies the number of cluster centers. After that, the data (clustered) are classified utilizing RNN.

3.4.2 Recurrent neural network algorithm

Subsequent to optimizing the features, the RNN properly classify the attacking as well as non-attacking labels. This is similar to that of the customary neural network except for one thing i.e., it is capable of remembering the entire information concerning the calculation by means of adding a memory state to the neurons. The same parameters are only used in all the input since the same task is only being lessening the intricacy of augmenting parameters and memorizing every preceding output by rendering every output as input to the subsequent hidden layer.

2. Thus, these '3' layers can well be united into a sole recurrent layer in order that the weights along with the bias of the entire hidden layers are the same. The RNN design is presented in Fig. 2.

The training and testing process is performed to classify the attacks. Already, the RNN is trained to do classification and the steps that are involved in RNN training is below,

1. A solo time step of the input is rendered to the network.
2. After that, compute its current state utilizing a collection of current input together with the preceding state. The equation for computing the current state is provided as,

$$h_t = f_t(h_{t-1}, i_t) \quad (9)$$

where, h_t is current state, h_{t-1} is previous states, and i_t is input state.

The current state h_t is activated by using Eq. (10),

$$h_t = \tanh(W_{hh} h_{t-1} + W_{ih} i_t) \quad (10)$$

where, W_{hh} is recurrent neuron's weight, and W_{ih} is input neuron's weight.

3. The current h_t changes to h_{t-1} for the succeeding time-step.
4. Several time-steps as possible can well be done in reference to the issue and link the information as of all done on the entire inputs or hidden layers (h) for generating the output. This lessens the parameters' complexity, not like other neural networks. The RNN functions as,

1. RNN transmutes the autonomous activations into dependent activations by means of rendering the same weights along with biases to every layer, therefore,

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 1, 2024

the preceding states.

5. Once the entire time steps are finished, the last current state is employed to compute the output, which is specified as,

$$o_t \quad ¼ \quad W_{ho}h_t \quad ð11Þ$$

wherein, o_t is output, and W_{ho} is weight of the output layer.

6. After that, the output is contrasted with the actual output (target output) and the error is produced.

$$\text{error} \quad ¼ \quad o_A \quad - \quad o_t \quad ð12Þ$$

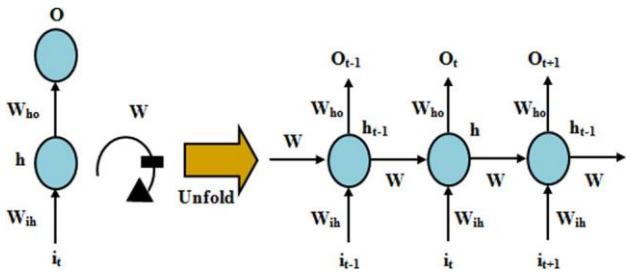

Fig. 2 Architecture of RNN

7. Then, the error is back-propagated towards the network for the updation of the weights, and thus, the network (RNN) is trained.

The testing of the data (clustered) values centered upon the training outcomes is performed. The last classification outcome encompasses '2' classes: (i) normal data, (ii) attack data. For the former one, encryption is performed for ameliorating the information's SL. And the Caesar cipher along with MECC is employed to ameliorate the SL, which is elucidated in the sections below.

3.5 Ciphering

The conversion of plaintext to ciphertext is termed the ciphering. And here, the Caesar Cipher (C_c) method is employed for ciphering. This is a sort of substitution cipher wherein every letter in the plain-text is 'shifted' to a particular number of places along the alphabet. For instance, with a shift of one, A will be swapped by B, B by C, and so forth. The C_c can well be mathematically written as:

$$C_c ¼ ðD_n þ sÞ \bmod 26 \quad ð13Þ$$

wherein, D_n denoted as original data (normal data) and s is the number of letters shift which can take the value from 1 to 25. This technique could only render minimal security to the data and also the frequency of the letter pattern render a big hint to decipher the complete data. Thus, the MECC algorithm is also deliberated to ameliorate security.

3.6 Encryption using MECC

Encryption is done to ensure only the certified person can access the data and not the unauthorized one. This is done by encoding the data. And in the proposed work, the C_c is encrypted by the MECC for enhancing data security. The meticulous elucidation regarding the MECC is depicted in the section below.

3.6.1 Modified ECC

A scheme is represented by elliptic-curve cryptography (ECC) for public-key cryptography. This is centered on a curve with specific base points employing a prime number function. It is utilized as a maximal limit. Besides, the ECC is hard to implement which increases the implementation errors' probability, expands the encrypted message's size, which in turn lessens the algorithm's security. Therefore, MECC is developed for enhancing security. Only the public along with the private key is formed in the ECC for encoding information whilst the security key is also produced for ameliorating the system's security in MECC.

This security key is incorporated with the encryption equation along with deducted from the decryption equation. Hence, the complication of the '2' phases is augmented. If the complication of encryption along with decryption is increased, then it will be extremely hard to identify the original data. It automatically improves the data's SL. The elliptic curve is specified by an equation of '2' variables with coefficients. The set of points are satisfied by the elliptic curve over real numbers, which should fulfill the Eq. (14) also known as the Weierstrass equation. It is given as,

$$q^2 ¼ y^3 þ ay þ b \quad ð14Þ$$

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 1, 2024

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 1, 2024

where, q; y; a and b signifies real numbers, applying disparate sets of values for a and b. The encryption's strength in a cryptographic method is always dependent on the key generation mechanism. In the proposed work, '3' kinds of keys are required to be generated. And the primary step is to produce the public key (k_{pu}) for data encryption. The next step is to create the private key (k_{pr}) for data decryption. At last, the generation of the secret key from the k_{pu}, k_{pr}, along with the point on the curve is done. Regard that a point b_c as a base point on the curve. Choose a random number within [0, n − 1] to create a k_{pr}. The k_{pu} is produced as,

$$k_{pu} = k_{pr} \omega b_c: \quad (15)$$

Then, the k_s is generated by summing the k_{pu}, k_{pr} and b_c which is specified as,

$$k_s = k_{pu}; k_{pr}; b_c : \quad (16)$$

After the key generation, the C_c values are encrypted. '2' cipher texts are contained by the encrypted data that aremathematically denoted as,

$$C_{t1} = (r_1 \omega b_c) + k_s \quad (17)$$

$$C_{t2} = C_c + (r_1 \omega k_{pu}) + k_s \quad (18)$$

wherein, C_{t1} and C_{t2} denotes cipher text1 along with cipher text2, r_1 signifies the arbitrary number in the range [1, n − 1]. From the decryption method, the original data is attained. The converse of encryption is called decryption. The created secret key is subtracted from normal decryp- tion equation in the decryption. It is formulated as,

$$C_c = (C_{t2} - k_{pr}) \omega C_{t1} - k_s \quad (19)$$

where, C_c implies the ciphered data, which is decrypted by inversing the ciphering method. It is called the deciphering, which is performed by the substrate s from Eq. (16) that is the original one. Maximum security is offered by this encryption to the data.

1 Result and discussion

Here, the performance examination of the proposed technique that is utilized in attack detection and prevention is analogized with the existing techniques, like ANN andECC concerning precision, accuracy, encryption, decryp- tion time, recall, SL, *F*-measure, and also memory usage on encryption along with decryption. The proposed work is effectively applied in the platform of JAVA with the following configuration.

1.1 Performance analysis of NK-RNN

The proposed one uses the NK-RNN classifier for the classification of AD on SCADA. The proposed NK-RNN classifier is analogized with existing ANN concerning *F*-measure, precision, accuracy, and recall. The proposed and existent techniques' performance comparison are delineated in Table 1

The NK-RNN classifier's performance with that of the existing ANN is exhibited in Table 1. The classifiers are contrasted with regard to metrics (*F*-measure, recall, precision, together with accuracy). For each performance investigation, the NK-RNN achieves superior performance. The diagrammatic depiction of Table 1 is specified below. Figure 3 exemplifies the performance comparison of NK-RNN with that of the ANN concerning precision. A disparate number of sensor values (SV) is chosen for performance assessment. Aimed at 1000 SV, NK-RNN as well

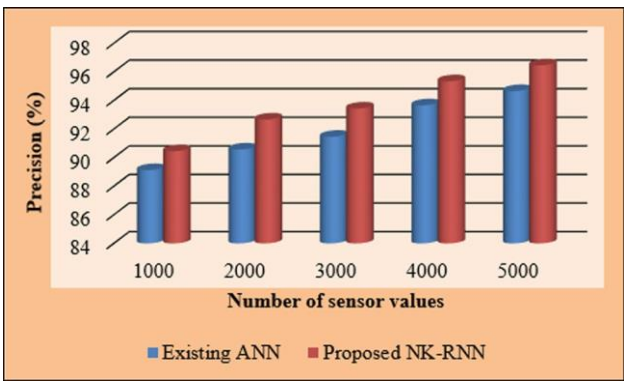

Fig. 3 Comparative analysis proposed and existing classification techniques in terms of precision

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 1, 2024

as ANN achieves 90.45% for precision. For the remainingSV, NK-RNN achieves the same precision values as ANN.Recall performance of the NK-RNN with that of the prevailing work is examined in Fig. 4. Here, the proposedwork renders 93.112% for recall in respect of 1000 SV,while the existent method renders 92.1123%, which islower than the proposed work. Likewise, for 2000, 3000,4000 as well as 5000, the proposed work renders93.3534%, 93.232432%, 94.83232% as well as 96.83223%, which are greater compared to the ANN recall values. Thus, it can well be stated that the proposed NK-RNN encompasses a better performance than the ANN.

Figure 5 exhibits the proposed NK-RNN's performance with that of the ANN concerning *F*-measure. These were contrasted centered upon the SV. Aimed at 5000 SV, the proposed achieves the highest *F*-measure of 96.854%,

Table 1 Performance comparison of proposed NK-RNN and Existing ANN in terms of (a) precision, recall and (b) *F*-measure, accuracy

(a)

Number of sensor data	Precision (%)		Recall (%)	
	Existing ANN	Proposed NK-RNN	Existing ANN	Proposed NK-RNN
1000	89.1245	90.4534	92.1223	93.112
2000	90.5678	92.6644	91.8855	93.3534
3000	91.4578	93.43545	92.2333	93.232432
4000	93.6578	95.34223	93.88775	94.83232
5000	94.6578	96.4555	95.8876	96.83223

(b)

Number of sensor data	*F*-measure (%)		Accuracy (%)	
	Existing ANN	Proposed NK-RNN	Existing ANN	Proposed NK-RNN
1000	90.23445	92.0455	82.12	84.776
2000	91.11234	93.14545	83.76	86.88
3000	92.23345	93.8875	86.92	90.344
4000	93.56674	95.1435	87.67556	91.776
5000	95.88754	96.854	91.678	93.66543

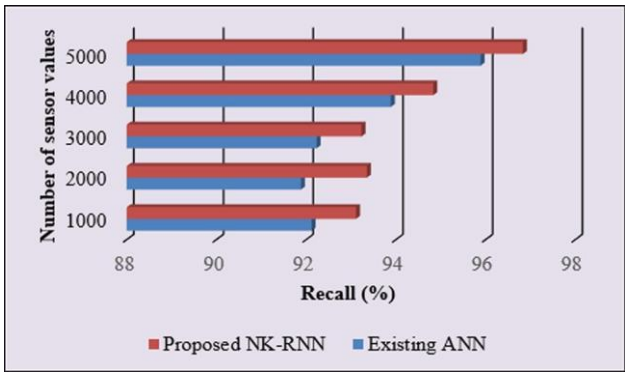

Fig. 4 Comparative analysis proposed and existing classification techniques in terms of recall

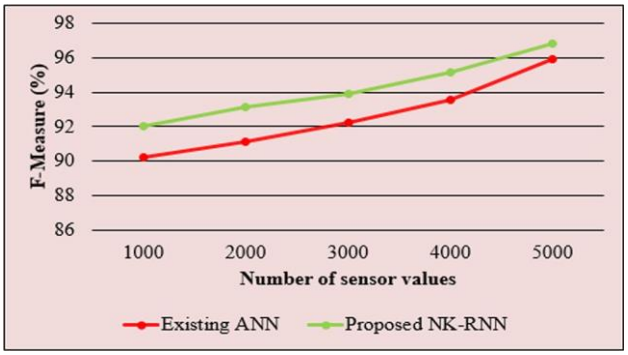

Fig. 5 Comparative analysis proposed and existing classification techniques in terms of *F*-measure

whereas, the ANN achieves 95.88754%, which is 0.96646% low to the proposed one. Besides, for the remaining SV, the proposed NK-RNN has the greatest *F*-measure value. Thus, it is inferred that the proposed NK-RNN gave superior outcomes to the ANN.

Figure 6 shows the proposed in addition to the existent method's comparison graph grounded on accuracy. As of Fig. 6, it can well be stated that the ANN showed less

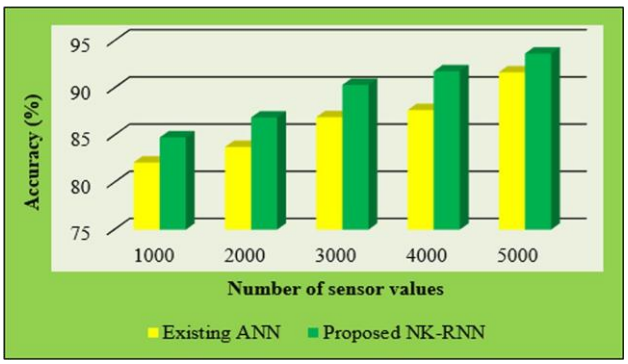

Fig. 6 Comparative analysis proposed and existing classification techniques in terms of accuracy

performance for all SV. Aimed at 1000 SV, the proposed NK-RNN achieved 84.774% of accuracy, while ANN achieved 82.12%, which is below the proposed one. Aimedat 2000, 3000, 4000 and 5000 SV, the NK-RNN achieved 86.88%, 90.334%, 91.776%, and 93.66543% of accuracy, correspondingly, which is better than the existent one. Likewise, the accuracy differs for the rest of the SV. Therefore, the proposed NK-RNN gave ameliorated per- formance than ANN. The suggested NK-RNN classifier outperforms the existing ANN. This is demonstrated by theresults for 1000 sensor values, which show that the pro- posed method achieved a precision of 90.4534%, recall of 93.3534%, *F*-measure of 90.23445%, and accuracy of 84.776%, while the existing ANN achieved values thatwere lower than the above-mentioned values. This is the case for other numbers of sensor values also. This better performance is mainly due to the fact that a normalized weight value will be applied to the initialization of the centroid in the existing k-means algorithm in NK-RNN to avoid unwanted outliers. Apart from this, the performance of the attack detection accuracy is enhanced by EHO via ignoring the irrelevant features from the incoming data.

1.2 Performance analysis of MECC

The MECC is utilized for security augmentation. The proposed MECC is weighed against existing ECC con- cerning encryption time, decryption time, SL, along with Memory utilization on encryption and also decryption which was exhibited in Figs. 7, 8, and 9.

1.2.1 Encryption time and decryption time

Figure 7 depicts the performance comparison of the pro- posed and the prevailing methods. The encryption execu- tion time of the proposed one is analogized with the ECC in figure (a). For '1000' SV, 26520 ms is taken by the pro- posed one to encrypt the data whilst 30,095 ms is taken by the existent ECC for their execution. The encryption time augments slowly when the SV increases. However, lesser time is consumed by the proposed work for every execu- tion when analogized to other existent methods. For the number of SV, the decryption time is plotted in Fig. (b). Figure (b) obviously exhibits that less decryption time is offered by the proposed one for every SV when analogizedto the prevalent one. So, it is evident that superior perfor- mance is possessed by the proposed one when contrasted toECC.

1.2.2 Security level

Figure 8 exhibits MECC and ECC's security level. The SL is the measure of strength that a cryptographic primitive,

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 1, 2024

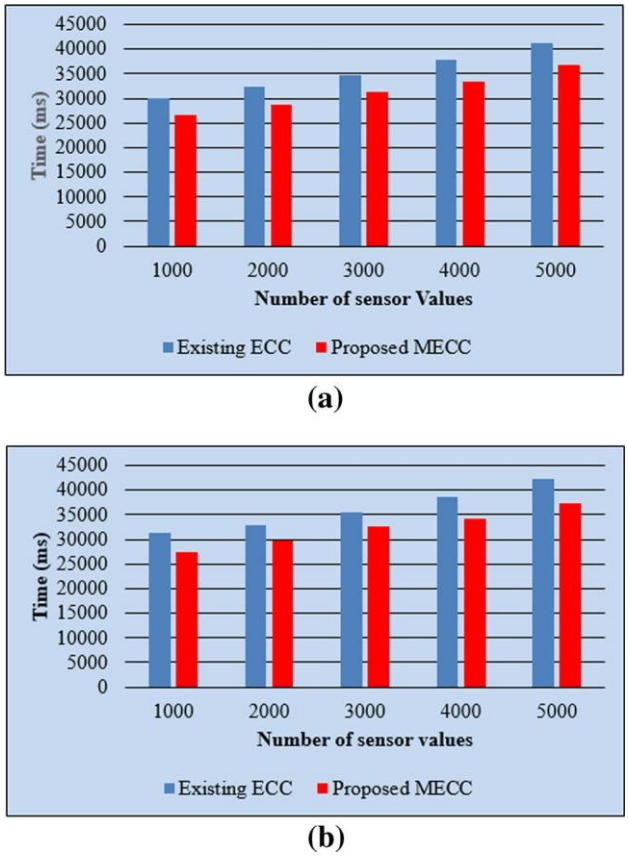

Fig. 7 Comparative analysis of proposed MECC with existing ECC in terms of a encryption time and b decryption time

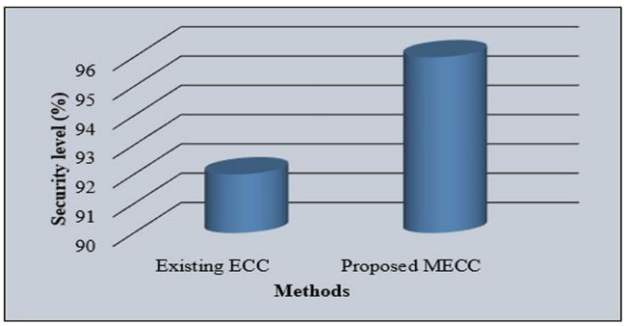

Fig. 8 SL of proposed MECC with existing ECC

say a cipher or hash function, obtains. The MECC obtains 96% of SL; whilst, ECC acquires only 92% that is lower than MECC. The improved result is mainly due to the fact that in ECC, only the public and private keys are generated for encoding data; however, in MECC, the security key is also created for improving the system's security. The encryption equation includes this security key, which is deducted from the decryption equation. As a result, the two phases become more complicated. It will be extremely hard

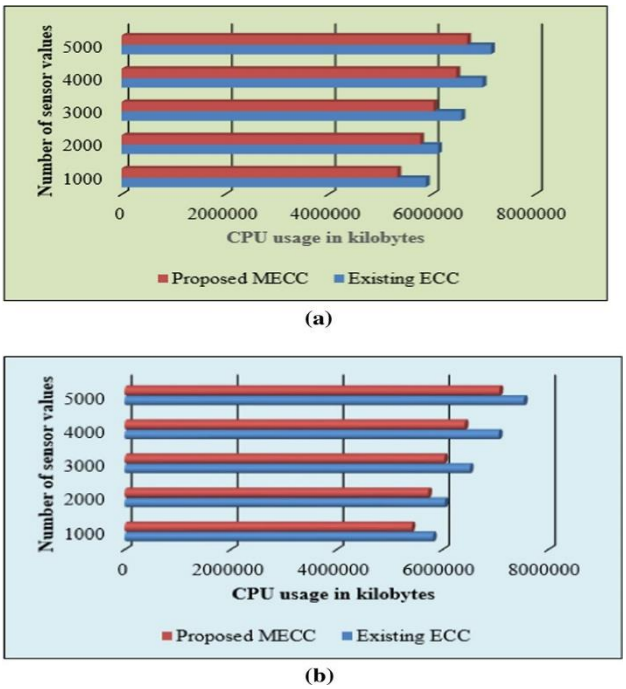

Fig. 9 Comparative analysis of proposed MECC with existing ECC in terms of a memory usage on encryption and b memory usage on decryption

to discern the original data if the complexity of encryption and decryption is increased. It automatically improves the data's SL. Thus, it can well be stated that higher perfor- mance is possessed by MECC when analogized to ECC concerning encryption, decryption time, and SL.

1.2.3 Memory usage on encryption and decryption

Figure 9 shows the assessment of the proposed MECC's performance with the ECC regarding memory usage on encryption in addition to decryption. The CPU utilization size is increased gradually when the SV increased. In Fig. (a), the memory usage of the proposed one on encryption is weighted against existing ECC. In the proposed MECC algorithm, the CPU uses 533,3457 kb memory for 1000 SV during the encryption process whereas, in the existing technique, the memory occupied by the CPU is 599,965 kb, which is bigger than the proposed one. For all SV, the proposed work uses less memory size compared with the existing ECC. And also in Fig. (b), for 3000 SV, the proposed method attained the value of 6,033,761 while the existing one acquired the value of 6,520,500. Figure (b) evidently shows that the proposed

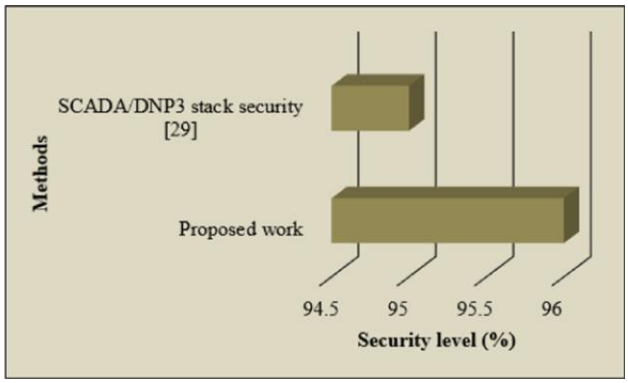

Fig. 10 Comparative analysis of proposed MECC with SCADA/DNP3 stack security regarding security level

one uses less memory space on decryption for every number of SV when weighted against the existing one. The proposed method showed better performance regarding memory usage, encryption time, and decryption time which is mainly because of caeser cipher usage. Caeser cipher provides the advantages such as the use of only a short key in the entire process and requiring few computing resources. To show the efficiency of the proposed work, it is analogized with one of the existing works in the literature named SCADA/DNP3 stack security (Shahzad et al. 2015a, b) based on the security level. The comparison is shown in the following Fig. 10.

Figure 10 exhibits MECC and ECC's security level. The SL is the measure of strength that a cryptographic primitive (a cipher or hash function) attains. The MECC attains 96% of SL while SCADA/DNP3 stack security attains only 95% that is lower than MECC. This shows the efficiency of the presented method for providing security.

2 Conclusion

A novel model is proposed for feature classification of the attack sorts in a SCADA with superior performance. The EHO, in the proposed work, enhance the accuracy by selecting the best features as of the feature dataset. Then, the proposed NK-RNN classifier classifies the optimized feature to normal data or attack data. The chief aim of this work is to precisely detect the intrusion in the SCADA network, security enhancement, along with training time reduction. In experiments, the performance outcomes of existent along with proposed methods are estimated concerning accuracy, precision, *F*-measure, recall, encryption and decryption time along with SL. As of this analysis, it is unmistakably stated that the proposed work shows better accuracy when analogized to the existing ones and attains 96% SL, which is 4% higher than the existing techniques.

Acknowledgements We thank the anonymous referees for their useful suggestions.

Author contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Mr. YJ, Dr. PJ. The first draft of the manuscript was written by Mr. YJ and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding This work has no funding resource.

Availability of data and materials Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Consent of publication Not applicable.

References

Abusafat F, Pereira T, Santos H (2018) Proposing a behavior-based IDS model for IoT environment. In: International Journal of European symposium on systems analysis and design. Springer, Cham, pp 114–134

Almalawi A, Fahad A, Tari Z, Alamri A, AlGhamdi R, Zomaya AY (2015) An efficient data-driven clustering technique to detect attacks in SCADA systems. IEEE Trans Inf Forensics Secur 11(5):893–906

Ambusaidi MA, He X, Nanda P, Tan Z (2016) Building an intrusion detection system using a filter-based feature selection algorithm. IEEE Trans Comput 65(10):2986–2998

Amin S, Litrico X, Sastry SS, Bayen AM (2012) Cyber security of water SCADA systems—part II attack detection using enhanced hydrodynamic models. IEEE Trans Control Syst Technol 21(5):1679–1693

Cherdantseva Y, Burnap P, Blyth A, Eden P, Jones K, Soulsby H, Stoddart K (2016) A review of cyber security risk assessment methods for SCADA systems. Comput Secur 56:1–27

Enescu FM, Bizon N (2017) SCADA applications for electric power system. Reactive power control in AC power systems, 1st edn. Springer, Cham, pp 561–609

Finogeev AG, Finogeev AA (2017) Information attacks and security in wireless sensor networks of industrial SCADA systems. J Ind Inf Integr 5:6–16

Ghosh S, Sampalli S (2019) A survey of security in SCADA networks current issues and future challenges. IEEE Access 7:135812–135831

Goldenberg N, Wool A (2013) Accurate modeling of Modbus/TCP for intrusion detection in SCADA systems. Int J Crit Infrastruct Prot 6(2):63–75

Hassan WH (2019) Current research on Internet of Things (IoT) security: a survey. Comput Netw 148:283–294

Hemdan EE-D, Manjaiah DH (2018) Cybercrimes investigation and intrusion detection in internet of things based on data science methods. In: Cognitive computing for big data systems over IoT. Springer, Cham, pp 39–62

Kalech M (2019) Cyber-attack detection in SCADA systems using temporal pattern recognition techniques. Comput Secur84:225–238

Krishnan Sadhasivan D, Balasubramanian K (2017) A novel LWCSO-PKM-based feature optimization and classification of attack types in SCADA network. Arab J Sci Eng 42(8):3435–3449

Li W, Xie L, Deng Z, Wang Z (2016) False sequential logic attack onSCADA system and its physical impact analysis. Comput Secur 58:149–159

Lin C-Y, Nadjm-Tehrani S, Asplund M (2017) Timing-based anomaly detection in SCADA networks. In: Internationalconference on critical information infrastructures security.Springer, Cham, pp 48–59

Lin H, Slagell A, Kalbarczyk ZT, Sauer PW, Iyer RK (2018) Runtimesemantic security analysis to detect and mitigate control-related attacks in power grids. IEEE Trans Smart Grid 9(1):163–178

Lu K-D, Zeng G-Q, Luo X, Weng J, Luo W, Wu Y (2021) Evolutionary deep belief network for cyber-attack detection in industrial automation and control system. IEEE Trans Ind Inform17(11):7618–7627

McEvoy TR, Wolthusen SD (2011) Defeating node based attacks on SCADA systems using probabilistic packet observation. In: International workshop on critical information infrastructuressecurity. Springer, Berlin, pp 70–80

Nazir S, Patel S, Patel D (2017) Assessing and augmenting SCADAcyber security a survey of techniques. Comput Secur 70:436–454 Samdarshi R, Sinha N, Tripathi P (2015) A triple layer intrusiondetection system for SCADA security of electric utility. In: Annual IEEE India conference (INDICON), 17–20 December,
New Delhi, India, pp 1–5

Shahzad A, Udagepola KP, Lee Y, Park S, Lee M (2015a) Thesensors connectivity within SCADA automation environment
and new trends for security development during multicasting routing transmission. Int J Distrib Sens Netw. https://doi.org/10. 1155/2015/738687

Shahzad A, Xiong N, Irfan M, Lee M, Hussain S, Khaltar B (2015b) A SCADA intermediate simulation platform to enhance the system security. In: 17th International conference on advanced communication technology (ICACT), 1–3 july, PyeongChang, Korea (South), pp 368–373

Shitharth S (2017) An enhanced optimization based algorithm forintrusion detection in SCADA network. Comput Secur 70:16–26 Shitharth S, Sangeetha K, Praveen Kumar B (2020) Integratedprobabilistic relevancy classification (PRC) scheme for intrusion detection in SCADA network. In: Design frameworks for wireless networks. Springer, Singapore, pp 41–63

Shlomo A, Kalech M, Moskovitch R (2020) Temporal pattern-based malicious activity detection in SCADA systems. Comput Secur. https://doi.org/10.1016/j.cose.2020.102153

Upadhyay D, Sampalli S (2020) SCADA (Supervisory Control and Data Acquisition) systems vulnerability assessment and security recommendations. Comput Secur 89:101666

Yang Y, McLaughlin K, Sezer S, Littler T, Im EG, Pranggono B, Wang HF (2014) Multiattribute SCADA-specific intrusion detection system for power networks. IEEE Trans Power Deliv 29(3):1092–1102

Yılmaz EN, Gönen S (2018) Attack detection/prevention system against cyber attack in industrial control systems. Comput Secur 77:94–105