Computers, Materials & Continua
DOI:10.32604/cmc.2020.013910
Article

Tech Science Press

A Real-Time Sequential Deep Extreme Learning Machine Cybersecurity Intrusion Detection System

Amir Haider[1], Muhammad Adnan Khan[2], Abdur Rehman[3], Muhib Ur Rahman[4] and Hyung Seok Kim[1,*]

[1]Department of Intelligent Mechatronics Engineering, Sejong University, Seoul, 05006, Korea
[2]Department of Computer Science, Lahore Garrison University, Lahore, 54000, Pakistan
[3]School of Computer Science, National College of Business Administration & Economics, Lahore, 54000, Pakistan
[4]Department of Electrical Engineering, Polytechnique Montreal, Montreal, QC H3T 1J4, Canada
*Corresponding Author: Hyung Seok Kim. Email: hyungkim@sejong.ac.kr
Received: 26 August 2020; Accepted: 28 September 2020

Abstract: In recent years, cybersecurity has attracted significant interest due to the rapid growth of the Internet of Things (IoT) and the widespread development of computer infrastructure and systems. It is thus becoming particularly necessary to identify cyber-attacks or irregularities in the system and develop an efficient intrusion detection framework that is integral to security. Researchers have worked on developing intrusion detection models that depend on machine learning (ML) methods to address these security problems. An intelligent intrusion detection device powered by data can exploit artificial intelligence (AI), and especially ML, techniques. Accordingly, we propose in this article an intrusion detection model based on a Real-Time Sequential Deep Extreme Learning Machine Cybersecurity Intrusion Detection System (RTS-DELM-CSIDS) security model. The proposed model initially determines the rating of security aspects contributing to their significance and then develops a comprehensive intrusion detection framework focused on the essential characteristics. Furthermore, we investigated the feasibility of our proposed RTS-DELM-CSIDS framework by performing dataset evaluations and calculating accuracy parameters to validate. The experimental findings demonstrate that the RTS-DELM-CSIDS framework outperforms conventional algorithms. Furthermore, the proposed approach has not only research significance but also practical significance.

Keywords: Security; DELM; intrusion detection system; machine learning

1 Introduction

Across various areas, such as social networking, online banking, and web browsing, the number of Internet users worldwide continues to grow and reach new heights. The increase in Internet usage is associated with an increase in cyberattacks, which poses a danger to the cybersecurity of the organizations. The primary reason for this is the growth of the Internet of Things (IoT) [1]. Cyber threats can result in irretrievable network disruption and economic difficulties. Such cyber threats can also

accompany a denial-of-service (DoS), device ransomware, or inappropriate approach [2,3]. Data suggests that a previous ransomware attack triggered a massive loss to several companies involving banking, education, resources, and colleges [4].

Throughout today's web-enabled communities, the increasing availability of network access enhances cyber-related violations. This rise is incentivizing organizations, which are trying to prevent illegal activities, to resolve network protection concerns. Cybersecurity architecture comprises a network protection framework and a computer security system [5]. While numerous technologies, including firewalls and cryptography, are configured to manage cyber-attacks on the Internet, a new intrusion detection system (IDS) is likely to withstand cyber threats on a computer network [6]. Traditional methods, such as firewalls, cannot withstand threats effectively [7,8].

An IDS detects harmful cyber-attack actions on a system while tracking and reviewing everyday operations in a system and the information structure to recognize security vulnerabilities and challenges. The need for information security has recently increased, requiring safety from various forms of cyber-attacks. An IDS often aims to track, monitor, and classify unwanted device activities such as illegal entry, alteration, or disruption [9,10]. Identifying different types of cyber threats and irregularities in a system and creating an efficient IDS integral to ongoing network security to ensure the protection of a device are critical. Consequently, the key objective of an IDS is to identify various forms of unauthorized network interaction and application systems used for early detection. Intrusion monitoring devices capture and archive network data in a server that can be further used for an analytical review of the network.

An IDS can vary in classification, depending upon the nature of implementation. The primary forms of IDSs include host-based and network-based systems with specific computer systems linked to complex systems. A host-based IDS is focused on a separate device and monitors critical operating system records for unusual or harmful incidents, limited to detecting and analyzing harmful content [11]. In contrast, the program analyzes and monitors the network infrastructure in a network IDS for unauthorized activity. Furthermore, well-established variations focusing on the detection method include signature-based and anomaly-related detection, which the global technology research community has investigated for several years [12].

For detecting subsequent threats, conventional techniques use a process characterized by a signature-based IDS. For example, the system perceives a byte series in system activity, established forms, or malware clusters as a signature. The antivirus program uses these template styles as a signature to identify attacks by comparing them. These signature-based IDSs can capture identified threats efficiently; nevertheless, identifying new unseen threats that use a recognized signature where no mechanism is visible is challenging [13]. In contrast, an anomaly-based IDS analyses the system's activity and discovers correlations, generates a data-driven framework for monitoring typical behavior, and identifies exceptions in the event of abnormalities. The anomaly-based IDS have the largest advantage over a signature-based IDS for tracking actions to manipulate existing and unknown flaws or cyber-attacks. The anomaly-based IDS can also yield false detection concentrations recognizing previously unknown device activities as abnormalities. Therefore, an efficient detection method focused on machine learning (ML) is required to mitigate these problems, which is the motivation for this work.

Predictive protection models must be developed to examine various trends of cyber-attacks and forecast the risks that can be applied to create an intelligent IDS that leverages cybersecurity data. The nature of artificial intelligence (AI) approaches, which can learn from a security database, are suitable for this task [14]. Therefore, in this research, we minimize security issues and propose an efficient data-driven intrusion detection framework in the field of cybersecurity. In this article, we propose an intrusion detection framework focused on a Real-Time Sequential Deep Extreme Learning Machine Cybersecurity Intrusion Detection System (RTS-DELM-CSIDS), which mitigates the above problems.

A comprehensive IDS is required because conventional approaches use a signature-based methodology to identify unique arrangements. One approach is to use the latest technologies, such as RTS-DELM-CSIDS, to evaluate data flows and identify intrusions and attack patterns. Hence, it is essential to manage smart blockchain-based applications by developing robust and versatile algorithms to process such a vast quantity of data. ML, an AI framework, involves machines for training, reasoning, and behaving without human intervention. The goal of ML is to create an effective algorithm to take data from the input, generate a prediction, and change the outputs by statistical analysis. ML may process a significant amount of information and make decisions guided by evidence.

In our method, we initially considered the classification of protection features given their significance in the framework. We subsequently developed an RTS-DELM-CSIDS security framework grounded on the selected core characteristics. After the development of the framework using training security information, we authenticated the framework using assessment outcomes. This method is not only successful in forecasting precision for unknown datasets by eliminating redundancy in simulation but also mitigates the framework's computational complexity by limiting the dimensions of the feature when constructing the corresponding structure. The outline of the efforts of the research are as follows:

- Identify the significance of high-dimensional safety features in an RTS-DELM-CSIDS.
- Formulate an IDS focused on the RTS-DELM-CSIDS security system.
- Consider the rating of security protocols based on their significance and construct a simplified model based on the specified core characteristics.
- Develop the proposed RTS-DELM-CSIDS framework for detecting intrusions on anomaly-based ML approaches to counter and track new attacks.
- Implement an innovative strategy for updating the learning model based on the feedback in the proposed RTS-DELM-CSIDS framework to raise the number of false warnings over a period.
- Decrease the risk of repeated false alerts with identical data in an efficient manner.
- Evaluate the efficacy of the proposed RTS-DELM-CSIDS framework for cyber intrusion detection.

The remainder of the paper is organized as follows. Section 2 briefly describes the related work. Section 3 elaborates on the mathematical modeling of the proposed system model. Section 4 presents the simulation results and discussion. Finally, Section 5 concludes the contributions and achievements of this study.

2 Related Work

Intrusion monitoring devices capture and archive network data in a server that can be further used for an analytical review of the network. Network analysis tools are built to examine vast amounts of network traffic data and information such as routers, network servers, and adapters. The key advantages of smart cities are increased protection, wellness, higher-quality education and living standards, efficient use of resources, improved management of the environment and habitats, a sustainable economy, and more jobs. Whereas the central concept of smart cities has existed for roughly ten years, it has been predicted to change city life drastically since it was first introduced with the emergence of essential facilitators, such as the IoT. Gogoi et al. [15] conducted a research study reviewing numerous current solutions to ML and the potential to identify threats in information on Internet traffic using unsupervised and supervised teaching methods. Peddabachigari et al. [16] incorporate support vector machine (SVM) and decision tree architectures for malware detection. Abbas et al. [17] proposed that a smart city is a sophisticated society in which diverse elements, such as people, the environment, sovereignty, democracy, and the economy, are interconnected in an intelligent network.

Diverse data mining (DM) and ML methods are used to identify sequences of security incidents to develop quality judgments [6]. An intrusion detection program is usually used to recognize suspicious

cyber-attack actions on a system when tracking and analyzing the everyday processes in a network or operating device to identify safety vulnerabilities or attacks [18]. This framework considers an established signature and has seen widespread acceptance and business success recently. Its methodology monitors network activity and identifies behavioral trends for risks by examining the related security information.

The anomaly-based method, in contrast, has an advantage over the signature-based approach to track invisible threats, such as the potential to recognize hidden or zero-day attacks [10]. Reducing the false positive level of an IDS should be a significant concern [12]. Within the field of cybersecurity, a range of work has been conducted for the potential to detect and deter cyber threats or breaches. Signature-based network intrusion detection is a popular method used in the cyber-industry [13]. The key downside of the anomaly-based approach is that it could generate substantial false alarm levels because it can identify undiscovered system operations as anomalies. However, an efficient detection method focused on ML is required to mitigate such problems.

Ganin et al. [1] proposed a comprehensive explanation of the functions of intrusion detection methods for ML. Buczak et al. [10] investigated cybersecurity intrusion detection methods for ML and DM. ML applications have drawn significant attention. Namdev et al. [19] proposed a detailed description using ML emphasis on Internet traffic classification. Bkassiny et al. [20] researched several complex knowledge complications in cognitive-radio networks (CRNs) and analyzed present ML-based solutions. Methods to address problems in wireless sensor networks with ML were examined in [21]. Wang et al. [22] developed state-of-the-art techniques to formulate heterogeneous networks in AI and discussed future research problems. Klaine et al. [23] investigated a useful classification and comparison of ML systems and their explanations in mobile networks. Fadlullah et al. [24] explored the use of ML methods to enhance network traffic management. Hodo et al. [25] concentrated on ML-based IDSs. Zhou et al. [26] emphasized the use of ML and cognitive radiation expertise to improve wireless-to-network spectrum use and energy efficiency. Abadeh et al. [27] suggested a complementary genetic area analysis algorithm to identify disruptive actions.

In contrast to the previous studies, we present in this investigation an RTS-DELM-CSIDS security framework. The proposed framework first captures the classification of security issues based on their significance. It then constructs a generic architecture for detecting intrusions centered on the identified significant characteristics to address the known problems.

3 System Model

This section elaborates on the development of the proposed IDS using anomaly-based ML methods. The proposed RTS-DELM-CSIDS framework uses an innovative strategy to mitigate the number of fake alerts over the period. It is achieved by accessing human expert feedback and modifying the learning model based on that information. This approach efficiently decreases the risk of repeated false alerts with identical data. Nonetheless, the proposed RTS-DELM-CSIDS can consider all unsupervised and supervised learning methods. Consequently, in the training process, there is no need for label information. Given that the percentage of traffic segments is the only necessary prior information, the proposed approach supports identifying the correct labels for unlabeled details. It also suggests adjustments in instances where conventional approaches classify the training samples through human experts.

The suggested IDS offer a scheme for simplifying the assessment of human safety experts' decisions. Accordingly, the framework can identify irregularities based on predetermined observations. Therefore, with supervised feature learning, the program has the potential to spot human mistakes while marking the data and proposing corrections. Moreover, the framework can identify new traffic segments using a scoring scheme. The suggested intrusion detection solution offers a rapidly-updating framework that identifies the adaptability dilemma of existing approaches. One use of the proposed framework is to

upgrade the learning framework based on current information and novel forms of attacks with minimal computational cost.

Fig. 1 describes the proposed RTS-DELM-CSIDS methodology applied to the NSL-KDD dataset. Intrusion detection is performed by two modules:

- A labeled information collection trains a learning model in the preprocessing layer to detect attacks.
- A novel attack detection method detects new threats in specific periods after RTS-DELM-CSIDS implementation.

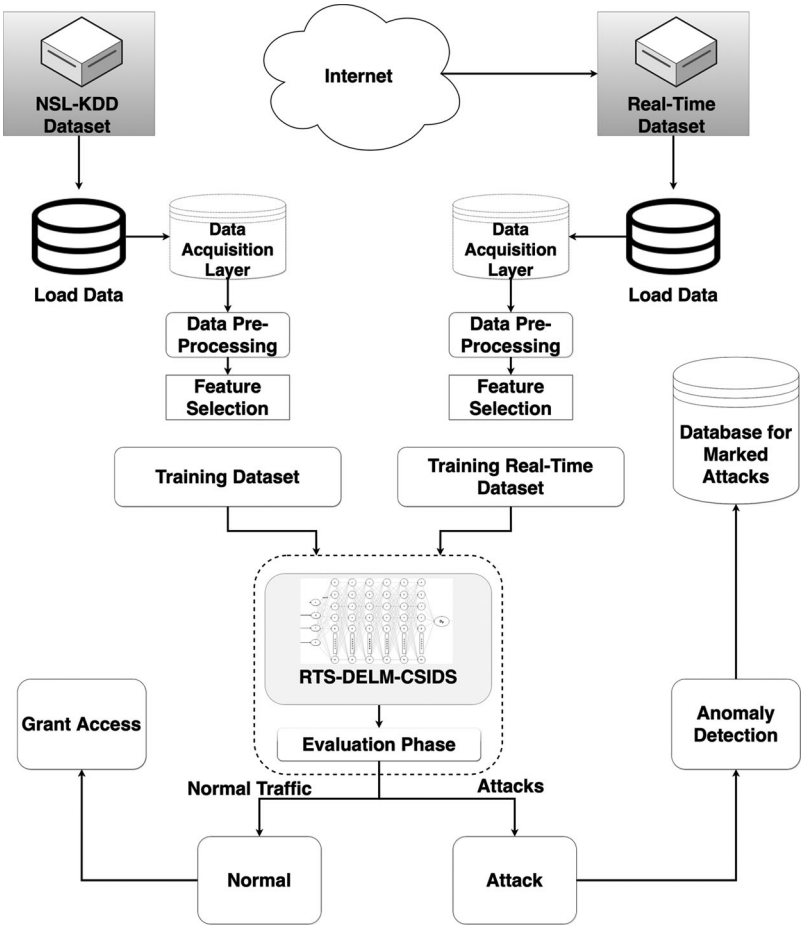

Figure 1: Proposed framework of RTS-DELM-CSIDS

3.1 Exploring Security Dataset

In this study, we use the publicly accessible intrusion dataset from Kaggle, which consists of two types: regular and attack [28]. Tab. 1 presents all the security protocols comprising their value categories. The NSL-KDD dataset is a revamped version of KDD 99, which has many improvements relative to the initial KDD 99 dataset [29]. The NSL-KDD dataset includes 41 features per record. Tab. 2 presents a full overview of the features.

Furthermore, each instance has a label indicating its traffic type—either regular or malicious traffic. The open attack groups for the NSL-KDD data belong to one of four major classes [28]: DoS, Remote to Local

(R2L), User-to-Root (U2R), and probe. All types of attacks include multiple sample attacks in the information collection. The complete list of threats is presented in Tab. 1

Table 1: NSL-KDD dataset attacks description [28]

DoS	Probe	R2L	U2R
back, land, neptune, pod, smurf, teardrop, processtable, udpstorm, mailbomb, apache2	ipsweep, nmap, saint, mscan, portsweep, satan	spy, warezclient, guesspassword, ftp_write, imap, multihop, named, phf, snmpgetattack, warezmaster, xlock, xsnoop, httptunnel, sendmail,	bufferoverflow, loadmodule, perl, snmpguess, sqlattack, xterm, rootkit, ps, worm

Table 2: Dataset structure [28]

Sr. No.	Features	Form of value	Sr. No.	Features	Form of value
1	duration	Integer	**22**	is_guest_login	Integer
2	protocol_type	Nominal	**23**	count	Integer
3	service	Nominal	**24**	srv_count	Integer
4	flag	Nominal	**25**	serror_rate	Float
5	src_bytes	Integer	**26**	srv_serror_rate	Float
6	dst_bytes	Integer	**27**	rerror _rate	Float
7	land	Integer	**28**	srv_rerror_rate	Float
8	wrong_fragment	Integer	**29**	same_srv_rate	Float
9	urgent	Integer	**30**	diff _srv_rate	Float
10	hot	Integer	**31**	srv_diff_host_rate	Float
11	num_failed_logins	Integer	**32**	dst_host_count	Float
12	root_shell	Integer	**33**	dst_host_srv_count	Float
13	num_compromised	Integer	**34**	dst_host_same_srv_rate	Float
14	roots_hell	Integer	**35**	dst_host_diff_srv_rate	Float
15	su_attempted	Integer	**36**	dst_host_same_src_port_rate	Float
16	num_root	Integer	**37**	dst_host_srv_diff_port_rate	Float
17	num_file_creations	Integer	**38**	ddst_host _serror_rate	Float
18	num_shells	Integer	**39**	dst-_host_srv_serror_rate	Float
19	num_access_files	Integer	**40**	dst_host_rerror_rate	Float
20	num_outbound_cmds	Integer	**41**	dst_host_srv_rerror_rate	Float
21	Is_host_login	Integer			

3.2 Deep Extreme Learning Machine (DELM)

Fig. 2 illustrates the Deep Extreme Learning Machine (DELM) framework, which has several hidden layers, hidden neurons, and activation functions of multiple types to produce an optimal cybersecurity

framework. The proposed framework consists of three layers: data collection, preprocessing, and application. The application layer includes two sub-layers: one for prediction and one for evaluation. The exploratory studies acquire actual information from sensors. The data collection layer uses the collected sensor data as inputs. Various cleanup processes of data and inspection methods are applied in the preprocessing layer to remove irregularities from the actual data. The application layer implements the DELM framework to optimize cybersecurity for any malicious or intrusive activity.

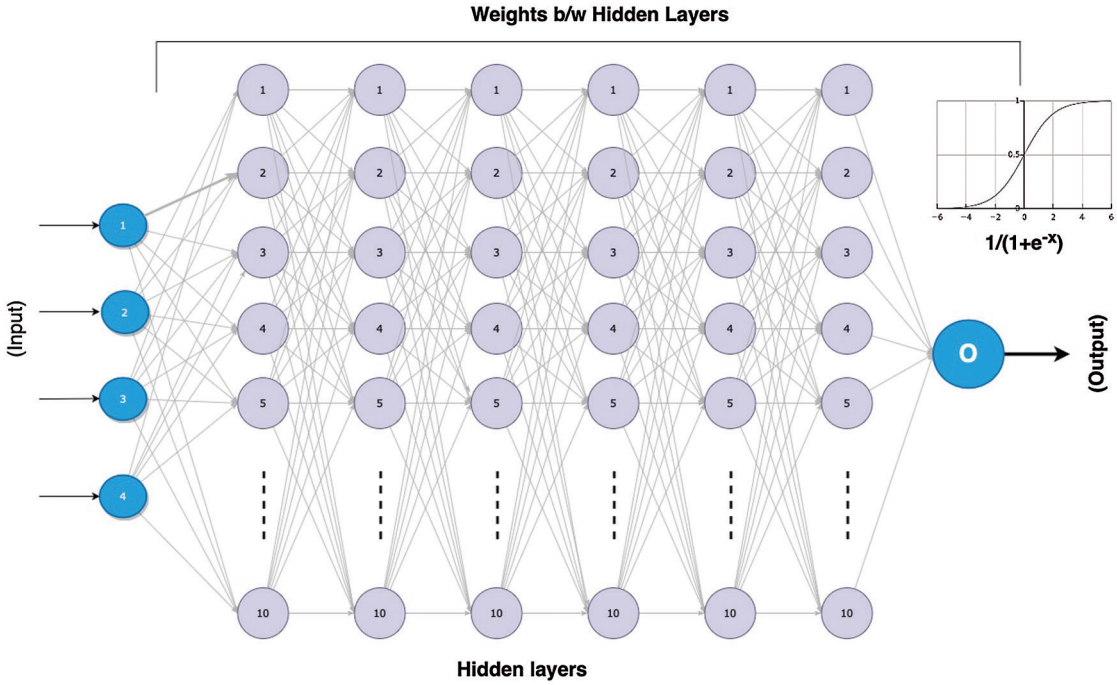

Figure 2: Systemic model of a DELM [30]

The DELM approach applies to multiple intrusion detection applications. A considerable number of sensor measurements is usually necessary to preserve the precision of the required detection. The DELM mitigates numerous network builder problems, such as network security and connectivity concerns. Given that sending and processing data consumes 80 percent of the network's energy, data reduction and feature extraction techniques can minimize processing and further prolong the network's lifespan. With high computational and memory demands, commonly used compression strategies may contribute to increased energy consumption. The DELM framework adjusts the data compression performance threshold within networks. Consequently, cybersecurity requires real-time networking approaches, including protection, schedules, tracking, clustering nodes, aggregating data, identification of faults, and data integrity. The DELM framework enhances the capacity of cybersecurity to respond to their surroundings' complex behavior.

A DELM can be configured in different fields to forecast health problems, energy consumption, and transport and traffic management [30,31]. Conventional artificial neural network (ANN) architectures require several measures and sluggish learning cycles and can overwrite the learning system [32]. The concept of an extreme learning machine is defined by [33], which is typically a feedforward neural network; this indicates that input passes only one direction over a set of layers. Nonetheless, we have also used the backpropagation approach in this predictive model during the training process, where data flows backward through the network. In backpropagation, the neural network adjusts weights to attain a high precision at a minimal error rate. The weights are constant during the validation stage, which imports the qualified model and predicts the real data. The DELM framework consists of an input layer, several hidden layers, and one output layer. After training of the system is complete, the trained model is

exported to the cloud for online use during the validation phase. Fig. 2 indicates the definition of the DELM in a hierarchical form, where m denotes nodes in the input layer, l denotes nodes in the hidden layer, and T denotes nodes in the output layer. The mean squared error (MSE) is observed in the evaluation layer to optimize cybersecurity.

Huang et al. [33] demonstrated an extreme learning machine (ELM) for the training of single-hidden-layer feedforward neural networks. In contrast, a DELM uses many current learning strategies (e.g., backpropagation) when arbitrarily initializing input weights; it only changes output weights in one iterative process without changing the input weights, thus offering a quick and comprehensive ability to learn. The DELM methodology operates as follows. Assume we have multiple hidden-layer feedforward neural networks with n hidden layer neurons and a training dataset of N records $(\ni_i, f_i)$, in which $(\ni_i \in S_d)$ and $f_i \in S_c$. The outcome of these multiple hidden layer feedforward neural networks is:

$$\sum_{j=1}^{n} \gamma_j ¥(p_j \ni_i + p_j),\ i \in [1, N] \tag{1}$$

where $\mathbf{p}_j$ and a_j are learning variables, γ_j is node output at weight j, and $¥ :\ \mathrm{S} \rightarrow \mathrm{S}$ is the activation function.

An ideal rapprochement of a multiple-hidden-layer feed forward neural network with zero error illustrates that with discrete intervals $\mathbf{p}_j$ and a_j there occurs γ_j such that;

$$\sum_{j=1}^{n} \gamma_j ¥(p_j \ni_i + a_j) = f_i,\quad i \in [1, N] \tag{2}$$

which can be represented as

$$\mathbf{Q}\gamma = \mathbf{F} \tag{3}$$

where

$$\mathbf{Q} = \begin{bmatrix} ¥(\mathbf{p}_1 \ni_1 + a_1) & & ¥(\mathbf{p_n} \ni_1 + a_n) \\ \vdots & \cdots & \vdots \\ ¥(\mathbf{p}_1 \ni_N + a_1) & & ¥(\mathbf{p_n} \ni_N + a_n) \end{bmatrix} \tag{4}$$

and

$$\gamma = \left(\gamma_1^T \ldots\ \gamma_n^T\right)^T,\ \mathbf{F} = \left(\mathrm{f}_1^T \ldots\ \mathrm{f}_N^T\right)^T. \tag{5}$$

When the number of observations is greater than the number of hidden-layer neurons, the outcome value weights can be determined using the following equation:

$$\gamma = \mathbf{Q}^{-1}\mathbf{F} \tag{6}$$

and $\mathbf{Q}^{-1}$ is the inverse of matrix $\mathbf{Q}$. A DELM is, therefore, a computationally economical system of study.

3.3 Real-Time Sequential Deep Extreme Learning Machine (RTS-DELM)

The DELM is a batch learning framework. We address instances in which information is received in sequential order by proposing a Real-Time Sequential DELM (RTS-DELM). Upon the creation of new information, the RTS-DELM upgrades the framework as a series of learning algorithms, without requiring previously trained data.

RTS-DELM comprises two initial and sequential stages of learning. Initially, the framework trains with N_0 different observations, and, based on these observations, the hidden layer performance matrix $\mathbf{Q}_0$ is developed. For DELM and RTS-DELM, N_0 should be equal to or less than the number of hidden layer neurons n, such that the level ($\mathbf{Q}_0$) is equivalent to or greater than n. The framework can be sequence-updated on the introduction of new information following the initial phase.

Suppose that $\{(\ni_i, \mathbf{f}_i)\}_{i=1}^{N_0}$ is the given training dataset at the earliest stage, $\mathbf{Q}_0$ and γ_0 are the hidden layer performance matrices, and the weights of the DELM output layer are as follows:

$$\mathbf{Q}_0 = \begin{bmatrix} ¥(\mathbf{p}_1 \ni_1 + a_1) & ¥(\mathbf{p_n} \ni_1 + a_n) \\ \vdots & \vdots \\ \cdots & \cdots \\ \vdots & \vdots \\ ¥(\mathbf{p}_1 \ni_{N_0} + a_1) & ¥(\mathbf{p_n} \ni_{N_0} + a_n) \end{bmatrix} \tag{7}$$

$$\gamma_0 = \mathbf{V}_0 \mathbf{Q}_0^T \mathbf{F}_0 \tag{8}$$

where $\mathbf{V}_0 = \left(\mathbf{f}_1^T\ \mathbf{Q}_0\right)^{-1}$ and $\mathbf{F}_0 = \left(\mathbf{f}_1^T \ldots \mathrm{f}_{N_0}^T\right)^T$.

After the initial stage, the creation of $(\mathrm{k}+1)^{\mathrm{th}}$ of records with N^{k+1} number of records may create the partially hidden layer performance matrix as

$$\mathbf{Q_{k+1}} = \begin{bmatrix} ¥(\mathbf{p}_1 \ni_{\left(\sum_{j=0}^{k} N_j\right)+1} + a_1) & ¥(\mathbf{p}_n \ni_{\left(\sum_{j=0}^{k} N_j\right)+1} + a_n) \\ \vdots & \vdots \\ \cdots & \cdots \\ \vdots & \vdots \\ ¥(\mathbf{p}_1 \ni_{\left(\sum_{j=0}^{k} N_j\right)+1} + a_1) & ¥(\mathbf{p}_n \ni_{\left(\sum_{j=0}^{k} N_j\right)+1} + a_n) \end{bmatrix} \tag{9}$$

and performance weights can be modified using the following equation:

$$\gamma_{\mathrm{k}+1} = \gamma_{\mathbf{k}} + \mathbf{V}_{\mathbf{k}+1} \mathbf{Q}_{\mathbf{k}+1}^{\mathbf{T}} (\mathbf{F}_{\mathbf{k}+1} - \mathbf{Q}_{\mathbf{k}+1} \gamma_{\mathbf{k}}) \tag{10}$$

with

$$\mathbf{V}_{\mathbf{k}+1} = \mathrm{V}_{\mathbf{k}} + \mathbf{V}_{\mathbf{k}} \mathbf{Q}_{\mathbf{k}+1}^{\mathbf{T}} \left(\mathbf{I} + \mathbf{Q}_{\mathbf{k}+1} \mathbf{V}_{\mathbf{k}} \mathbf{Q}_{\mathbf{k}+1}^{\mathbf{T}}\right)^{-1} \mathbf{Q}_{\mathbf{k}+1} \mathrm{V}_{\mathbf{k}} \tag{11}$$

The back propagation method involves weight configuration, feed forward propagation, backward error propagation, and a distinguish ability update. An activation function such as $\mathrm{g}(x)$ = sigmoid is present in the hidden layer on each neuron. This approach designs the sigmoid input function and the DELM hidden layer:

$$E = \frac{1}{2} \sum\nolimits_j \left(sj - wp_j \right) 2 \tag{12}$$

s_j = Desired output

wp_j = calculated output

Eq. (12) specifies a back propagation error, which can be measured by dividing the square sum from the required output by 2. The adjustment in weight is required to mitigate the common error. The weight change levels for the output layer are defined by Eq. (13) as:

$$\Delta \mathrm{H}_{i,j}^{l=6} \propto -\frac{\partial \mathrm{R}}{\partial \mathrm{H}^{l=6}} \tag{13}$$

$i = 1, 2, 3 \ldots 10$ $(no.of$ neurons$)$

and j = Layer of Output Value

Eq. (14) illustrate the weight update and how the biases occur between the hidden layer and inputs.

$$H_{i,n}^{l}(t) = H_{i,n}^{l}(t+1) + \lambda H_{i,j}^{l} \tag{14}$$

4 Results and Discussion

In this study, we implement an RTS-DELM methodology on a dataset [28]. The data were divided up randomly into 70% training (103,962 samples) and 30% validation (44,554 samples). Data were processed in advance to delete information irregularities and mitigate information from errors. The RTS-DELM-CSIDS attempts to identify any malicious action or intrusion in various hidden layers (including hidden neurons) and activation functions. Moreover, we evaluated a specific number of neurons in hidden layers in a network and also implemented numerous forms of active functions. In this investigation, we evaluated the RTS-DELM-CSIDS to accurately predict the efficiency of this system. We calculated the output with the counterpart algorithms of the RTS-DELM-CSIDS algorithm using multiple statistical measures, as described in Eq. (15) through Eq. (18).

Within this analysis, we assess the efficacy of the experiments conducted by the current RTS-DELM-CSIDS. The two widely used intrusion detection assessment measures are Detection Rate and False Positive Rate:

$$\text{Miss rate} = \frac{\sum_{k=0}^{2}\left(W_k/R_{j\neq k}\right)}{\sum_{k=0}^{2}(R_k)}, \text{where j} = 0, 1, 2 \tag{15}$$

$$\text{Accuracy} = \frac{\sum_{k=0}^{2}\left(W_k/R_k\right)}{\sum_{k=0}^{2}(R_k)} \tag{16}$$

$$\text{Detection Rate} = \frac{\text{Number of Intrusions Detection}}{\text{Total Number of Infused Intrusions}} \tag{17}$$

$$\text{False Positive Rate} = \frac{\text{Generic number of trends categorized as intrusions}}{\text{The overall number of standard patterns}} \tag{18}$$

In Eqs. (15), (16), W symbolizes the predictive outcome value of RTS-DELM-CSIDS and R expresses the real outcome value. W_0 and R_0 denote normal (no attack) in the predictive and actual outputs, respectively. W_1 and R_1 denote an attack in the predictive and actual outputs, respectively. The W_k/R_k indicates that projected and real outcome values are similar. Likewise, $W_k/R_{j\neq k}$ denotes error, where predictive and real outcome values vary.

Tab. 3 illustrates the proposed RTS-DELM-CSIDS for the prediction of intrusion detection during the training phase. A total of 103,962 samples were used during training, divided into 53,937 and

50,025 samples for normal and attacks, respectively; 52,198 samples belong to a normal class, in which no attack was found, and these samples were correctly predicted. Moreover, 1,739 samples were recorded as incorrect predictions, demonstrating that an attack was detected while, in reality, no attack occurred. Similarly, 50,025 samples were taken for attack found, for which the attack was correctly predicted for 47,835 samples; in contrast, 2,190 samples were incorrect predictions because they were found to be normal while, in reality, an attack occurred.

Table 3: Training of the proposed RTS-DELM-CSIDS security model during the prediction of network traffic

Proposed RTS-DELM-CSIDS Security Model (70% of Sample Data in Training)			
Type	Total No of Samples (N = 103962)	Result (Output) (W_0, W_1)	
Input	Expected Output (R_0, R_1)	W_0 (Normal)	W_1 (Attack)
	R_0 = 53937 Normal	52198	1739
	R_1 = 50025 Attack	2190	47835

Tab. 4 depicts the proposed RTS-DELM-CSIDS for the prediction of intrusion detection during the validation phase. A total of 44,554 samples were used during validation, which is further divided into 23,116 and 21,438 samples for normal and attacks, respectively. Moreover, 21,837 samples of the normal class were correctly predicted, while 1,279 samples exhibit incorrect prediction of an attack while, in reality, no attack occurred. Similarly, 21,438 samples were taken for attack found, for which the attack was correctly predicted for 19,478 samples; in contrast, 1,960 samples were incorrect predictions because they were found to be normal while, in reality, an attack occurred.

Table 4: Validation of the proposed RTS-DELM-CSIDS security model during the prediction of network traffic

Proposed RTS-DELM-CSIDS Security Model (30% of Sample Data in Validation)			
Type	Total No of Samples (N = 44554)	Result (Output) (W_0, W_1)	
Input	Expected Output (R_0, R_1)	W_0 (Normal)	W_1 (Attack)
	R_0 = 23116 Normal	21837	1279
	R_1 = 21438 Attack	1960	19478

Tab. 5 depicts the proposed RTS-DELM-CSIDS system model performance in terms of accuracy and miss rate during the training and validation phase. The proposed RTS-DELM-CSIDS system model had a 96.22% and 3.78% accuracy and miss rate during training, respectively. Moreover, during the validation phase, the proposed RTS-DELM-CSIDS system model had a 92.73% accuracy and a 7.27% miss rate.

Table 5: Performance evaluation of proposed RTS-DELM-CSIDS during validation and training

Process	Accuracy	Miss Rate
Training	96.22%	3.78%
Validation	92.73%	7.27%

We contrasted the efficiency of our solution to the current NSL-KDD dataset approaches. As illustrated in Fig. 3, with a lower error rate, the proposed system achieves considerably higher precision. The proposed RTS-DELM-CSIDS technique is superior to other models in terms of accuracy, such as a self-organizing map (SOM) [34], ANN-based IDS [35], and generative adversarial networks (GANs) [36]. The RTS-DELM-CSIDS outperformed the NSL-KDD dataset. The precision of the SOM [34] is much lower than other ML algorithms. In [34], the authors proposed a SOM, in which the investigators obtained 75.5% accuracy. In [35], the authors proposed ANN-based IDS, in which the investigators obtained 81.2% accuracy. In [36], the authors proposed GANs, in which the investigators obtained 86.5% accuracy. The proposed RTS-DELM-CSIDS framework accuracy is 96.22% and higher in terms of accuracy compared with the existing techniques. The proposed RTS-DELM-CSIDS framework outperforms current methods significantly based on statistical values. Thus, the proposed RTS-DELM-CSIDS framework is a viable option as a smart solution for network security.

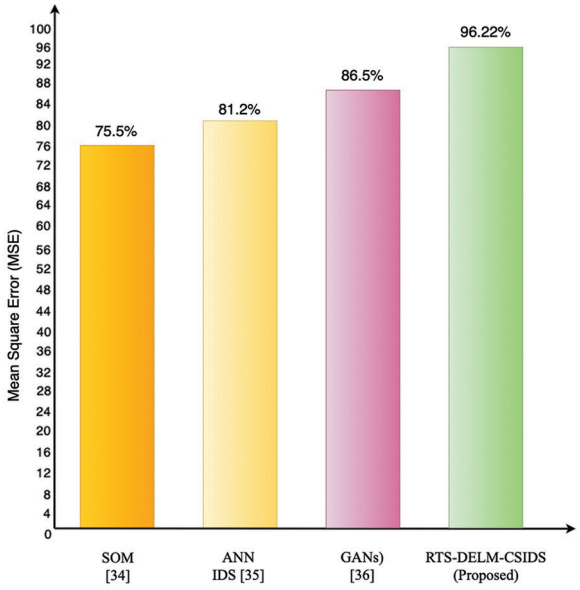

Figure 3: Comparison results of the proposed RTS-DELM-IDS with literature

5 Conclusions

In this article, we have proposed an RTS-DELM-CSIDS security model. We presented safety characteristics based on their values. Moreover, we developed a generalized intrusion detection model focused on the identified critical characteristics to ensure forecasting accuracy and efficiency for unknown datasets and reduce computational complexity. We then tested the efficacy of our RTS-DELM-CSIDS framework by performing a sequence of dataset experiments. Specific methodological approaches were used to determine the feasibility of the suggested approach. The measured data indicate that the proposed

RTS-DELM-CSIDS approach is much higher in accuracy compared with other algorithms. The proposed RTS-DELM-CSIDS security model produces impressive outcomes. The proposed technique exhibits 96.22% and 92.73% accuracy during training and validation, respectively. We also contrasted the findings of the RTS-DELM-CSIDS framework with many conventional mainstream approaches to evaluate the efficacy of the associated security framework.

Funding Statement: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (Nos. 2019R1A4A1023746, 2019R1F1A1060799) and Strengthening R&D Capability Program of Sejong University.

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

References

[1] A. A. Ganin, P. Quach, M. Panwar, Z. A. Collier, J. M. Keisler *et al.,* "Multicriteria decision framework for cybersecurity risk assessment and management," *Risk Analysis*, vol. 40, no. 1, pp. 183–199, 2017.

[2] N. Sun, J. Zhang, P. Rimba, S. Gao, L. Y. Zhang *et al.,* "Data-driven cybersecurity incident prediction: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1744–1772, 2019.

[3] I. Yaqoob, E. Ahmed, M. H. ur Rehman, A. I. A. Ahmed, M. A. Al-garadi *et al.,* "The rise of ransomware and emerging security challenges in the Internet of Things," *Computer Networks*, vol. 129, pp. 444–458, 2017.

[4] X. Qu, L. Yang, K. Guo, L. Ma, M. Sun *et al.,* "A survey on the development of self-organizing maps for unsupervised intrusion detection," *Mobile Networks & Applications*, vol. 24, no. 5, pp. 1–22, 2019.

[5] K. Yang, J. Ren, Y. Zhu and W. Zhang, "Active learning for wireless IoT intrusion detection," *IEEE Wireless Communications*, vol. 25, no. 6, pp. 19–25, 2018.

[6] N. Sultana, N. Chilamkurti, W. Peng and R. Alhadad, "Survey on SDN based network intrusion detection system using machine learning approaches," *Peer-to-Peer Networking and Applications*, vol. 12, no. 2, pp. 493–501, 2019.

[7] S. Mohammadi, H. Mirvaziri, M. G. Ahsaee and H. Karimipour, "Cyber intrusion detection by combined feature selection algorithm," *Journal of Information Security and Applications*, vol. 44, pp. 80–88, 2019.

[8] M. Tavallaee, N. Stakhanova and A. A. Ghorbani, "Toward credible evaluation of anomaly-based intrusion-detection methods," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 5, pp. 516–524, 2010.

[9] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li *et al.,* "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, 2018.

[10] L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[11] Y. Yao, Q. Fu, W. Yang, Y. Wang and C. Sheng, "An epidemic model of computer worms with time delay and variable infection rate," *Security & Communication Networks*, vol. 5, no. 2018, pp. 1–11, 2018.

[12] R. Sommer and V. Paxson, "Outside the closed world: on using machine learning for network intrusion detection," in *2010 IEEE Symposium on Security and Privacy*, Chicago, Illinois, USA, pp. 305–316, 2010.

[13] Q. Yan, F. R. Yu, Q. Gong and J. Li, "Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 602–622, 2016.

[14] H. Sarker, A. S. M. Kayes, S. Badsha, H. Alqahtani, P. Watters *et al.,* "Cybersecurity data science: An overview from machine learning perspective," *Journal of Big Data*, vol. 7, no. 1, pp. 1–29, 2020.

[15] P. Gogoi, B. Borah and D. K. Bhattacharyya, "Anomaly detection analysis of intrusion data using supervised and unsupervised approach," *Journal of Convergence Information Technology*, vol. 5, no. 1, pp. 95–110, 2010.

[16] S. Peddabachigari, A. Abraham, C. Grosan and J. Thomas, "Modeling intrusion detection system using hybrid intelligent systems," *Journal of Network and Computer Applications*, vol. 30, no. 1, pp. 114–132, 2007.

[17] S. Abbas, M. A. Khan, L. E. Falcon-Morales, A. Rehman, Y. Saeed *et al.,* "Modeling, simulation and optimization of power plant energy sustainability for IoT enabled smart cities empowered with deep extreme learning machine," *IEEE Access*, vol. 8, pp. 39982–39997, 2020.

[18] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer and B. D. Payne, "Evaluating computer intrusion detection systems: A survey of common practices," *ACM Computing Surveys*, vol. 48, no. 1, pp. 1–41, 2015.

[19] T. T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

[20] M. Bkassiny, Y. Li and S. K. Jayaweera, "A survey on machine- learning techniques in cognitive radios," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1136–1159, 2013.

[21] M. A. Alsheikh, S. Lin, D. Niyato and H. P. Tan, "Machine learning in wireless sensor networks: algorithms, strategies, and applications," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1996–2018, 2014.

[22] X. Wang, X. Li and V. C. M. Leung, "Artificial intelligence-based techniques for emerging heterogeneous network: State of the arts, opportunities, and challenges," *IEEE Access*, vol. 3, pp. 1379–1391, 2015.

[23] P. V. Klaine, M. A. Imran, O. Onireti and R. D. Souza, "A survey of machine learning techniques applied to self organizing cellular networks," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2392–2431, 2017.

[24] Z. M. Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi *et al.,* "State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2432–2455, 2017.

[25] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis and R. Atkinson, "Shallow and deep networks intrusion detection system: A taxonomy and survey," *ACM Survey*, 2017. [Online]. Available: http://arxiv.org/abs/1701.02145.

[26] X. Zhou, M. Sun, G. Y. Li and B. H. Juang, "Intelligent wireless communications enabled by cognitive radio and machine learning." *China Communications*, vol. 15, no. 12, pp. 16–18, 2018.

[27] M. S. Abadeh, J. Habibi, Z. Barzegar and M. Sergi, "A parallel genetic local search algorithm for intrusion detection in computer networks," *Engineering Applications of Artificial Intelligence*, vol. 20, no. 8, pp. 1058–1069, 2007.

[28] Kaggle, 2019. [Online]. Available: https://www.kaggle.com/hassan06/nslkdd.

[29] K. Siddique, Z. Akhtar, F. A. Khan and Y. Kim, "Kdd cup 99 data sets: A perspective on the role of data sets in network intrusion detection research," *Computer*, vol. 52, no. 2, pp. 41–51, 2019.

[30] M. A. Khan, S. Abbas, K. M. Khan, M. A. Al-ghamidi and A. Rehman, "Intelligent forecasting model of COVID-19 novel coronavirus outbreak empowered with deep extreme learning machine," *Computers, Materials & Continua*, vol. 64, no. 3, pp. 1329–1342, 2020.

[31] A. Rehman, A. Athar, M. A. Khan, S. Abbas, A. Fatima *et al.,* "Modelling, simulation, and optimization of diabetes type II prediction using deep extreme learning machine," *Journal of Ambient Intelligence and Smart Environments*, vol. 12, no. 2, pp. 125–138, 2020.

[32] C. Jiatan, "QAPSO-BP algorithm and its application in vibration fault diagnosis for a hydroelectric generating unit," *Journal of Vibration & Shock*, vol. 34, no. 23, pp. 177–181, 2015.

[33] G. Huang, D. Wang and Y. Lan, "Extreme learning machines: A survey," *International Journal of Machine Learning and Cybernetics*, vol. 2, no. 2, pp. 107–122, 2011.

[34] L. M. Ibrahim, D. T. Basheer and M. S. Mahmod, "A comparison study for intrusion database (Kdd99, Nsl-Kdd) based on self organization map (SOM) artificial neural network," *Journal of Engineering Science & Technology*, vol. 8, no. 1, pp. 107–119, 2013.

[35] B. Ingre and A. B. Yadav, "Performance analysis of NSL-KDD dataset using ANN," in *Int. Conf. on Signal Processing and Communication Engineering Systems*, Guntur, India, pp. 92–96, 2015.

[36] R. Alshinina and K. Elleithy, "A highly accurate machine learning approach for developing wireless sensor network middleware," in *2018 Wireless Telecommunications Symposium*, Phoenix, AZ, pp. 1–7, 2018.