

Machine Learning for Cloud Data Classification and Anomaly Intrusion Detection



Leila Megouache^{1*}, Abdelhafid Zitouni², Salheddine Sadouni³, Mahieddine Djoudi⁴

¹ Lire Laboratory, Geographical Sciences and Topography Department, Constantine 1 University, Constantine 25000, Algeria

² Lire Laboratory, Computer Science Department, Constantine 2 University, Constantine 25000, Algeria

³ LSIACIO Laboratory, Geographical Sciences and Topography Department, Constantine 1 University, Constantine 25000, Algeria

⁴ Techne, TECHNE Poitiers, Poitiers University, Poitiers 86073, France

Corresponding Author Email: megouache_leila@yahoo.fr

Copyright: ©2024 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/isi.290514>

ABSTRACT

Received: 20 December 2023

Revised: 11 September 2024

Accepted: 30 September 2024

Available online: 24 October 2024

Keywords:

artificial intelligence, extreme learning machine, cloud, intrusion detection system, security, k-means clustering

The sheer volume of applications, data and users working in the cloud creates an ecosystem far too large to protect against possible attacks. Several attack detection mechanisms have been proposed to minimize the risk of data loss backed up to the cloud. However, these techniques are not reliable enough to protect them; this is due to the reasons of scalability, distribution and resource limitations. As a result, Information Technology Security experts may feel powerless against the growing threats plaguing the cloud. For that, we provide a reliable way to detect attackers who want to break into cloud data. In our framework, we have no labels and no predefined classes on historical data, and we wish to identify similar models to form homogeneous groups from our observations. Then, we will use a k-means clustering algorithm to handle unlabelled data, and a combination approach of clustering and classification. We start with a k-means clustering algorithm for generating a labelled dataset from an unlabelled dataset. By harnessing the power of a labelled dataset, we can train the extreme learning machine classifier to become an exceptional tool for intrusion detection. By utilizing this resampling technique, we can generate additional data sets to significantly enhance the system's capability to identify and thwart attacks. The innovation of this approach stems from its integration of clustering and classification into a unified learning model. The cutting-edge framework has been successfully implemented on the renowned KDD99 dataset, producing impressive numerical results that not only affirm its exceptional accuracy but also highlight the significant time-saving advantages of this innovative approach.

1. INTRODUCTION

With the rapid expansion of the Internet of Things and digitization, various security incidents such as unauthorized access [1] and malware attack [2] have grown at an exponential rate in recent years.

Cloud computing is now widely adopted and used by large companies to take advantage of the delivery of applications, infrastructures and high storage capacity on the Internet. In today's dynamic cloud computing environment, the potential for multiple users to access a single server and seamlessly retrieve and update their data without the need for individual application licenses is truly revolutionary. This not only streamlines work processes but also opens up a world of possibilities for maximizing the benefits of cloud computing in our professional lives [3]. However, as the scale and intricacy of cloud operations continue to grow, it's crucial to emphasize the development of a robust security infrastructure to safeguard our valuable data and operations. For example, suppose an attack has occurred at the cloud level, all cloud resources will be permanently affected, and the quality of service will decrease. Therefore, the data protection of all

cloud users is damaged. For this, cloud service providers must protect their resources to maintain the quality of resources [4].

Although several solutions exist with adequate security measures for cloud applications, they are still insufficient compared to the speed of threats that emerge every day, and the spammers who keep on inspecting our operations. In addition, as cloud operations are shared between different actors, the interoperability factor also becomes a critical requirement [5]. For these reasons, we introduce machine learning for its speed and performance.

Machine learning (ML) is a game-changer that empowers computers to learn and adapt without the need for explicit programming [6]. It is a significantly large and growing field of artificial intelligence. Its purpose is to facilitate human tasks through its speed and automatic reasoning. Insecurity, machine learning is based on data analysis to uncover hidden patterns. This enables us to stay one step ahead by detecting malware in encrypted traffic, identifying internal threats, and even predicting the "bad neighborhoods" online to ensure a safer browsing experience. With machine learning, we can also safeguard sensitive data in the cloud by learning from and reacting to suspicious user behaviour [7]. In machine learning

security we often talk about three main types of attacks: poisoning, evasion and inference. In the case of poisoning, an attacker seeks to bias the behaviour of a model by modifying training data. We can take the well-known example of Microsoft Tay, a chatbot designed to interact on social networks with young Americans. It ended up appropriating the vocabulary of its speakers. With evasion, an attacker plays on the input data of the application to obtain a decision different from the one normally expected. And finally, in the inference case, an attacker successively tests different requests on the application to study its behavior [8]. There are currently several use cases of ML in the field of cyber security, such as fraud detection, vulnerability detection from predictive models, intrusion detection, static analyzes and the detection of infiltration of data. Rizal [2] delivered a comprehensive overview of the cutting-edge machine learning techniques used in identifying malicious URLs. The presentation delved into feature representation, learning algorithm development, and categorization in this crucial domain.

Existing solutions have primarily relied on distinguishing between "legitimate" and "malicious" connections within an IoT network. However, when it comes to public wireless networks, the attack detection system takes a different approach by utilizing a detection model that assesses the reputation and trustworthiness of each node [9, 10]. This system compares the reputation of each node against a predefined threshold to determine whether the node is trustworthy or should be flagged as a potential attacker. Or the innovative SVELTE system [11], a cutting-edge prototype to secure the Conkiti operating system. It includes a distributed mini firewall to respond to alerts. SVELTE is a hybrid system which has a centralized components and a distributed component between the nodes. It is installed both on the nodes and on the router which links the internal zone of the objects and the rest of the Internet. It is designed primarily to detect attacks on routing protocols.

So, our objective in this proposed work is to minimize the risk of intrusion at the cloud level by using probability laws and the K-means clustering algorithm for data segmentation and also to know how to use classification techniques to categorize the different attacks that may occur. Intrusion detection by the classification method only is increasingly used. However, building a system that utilizes classification and clustering techniques can significantly enhance the effectiveness of intrusion detection methods. Additionally, resampling enables the generation of new datasets, further improving the system's ability to identify and thwart potential attacks. The KDD 1999 intrusion detection dataset will play a role in our key to solving the problem and is the most widely used by researchers working in the security field. Then our contribution in this paper is to create a framework based on a combination of clustering and classifier to optimize the quality of our system and reduce false positives. Firstly k-means clustering is used to create a labeled dataset from an unlabeled dataset. The labeled dataset plays a crucial role in training the ELM classifier, which is the key to detecting and preventing intrusions. The experiments with the KDD99 dataset show a high quality of intrusion detection.

In this paper, we first discuss data security concepts and relevant problem-solving methods through intelligent decision-making in a distributed environment. We also make a brief discussion of different machine learning tasks in security. Second, we propose an extensible methodology to model user behavior from contextual information. Behaviors

follow a probabilistic procedure to filter out malicious operations. Finally, we try to improve data security by combining clustering and classification methods. The results of implementing this method are profound. The method has significantly enhanced the precision of filtering data stored in the cloud, thereby ensuring that only relevant and accurate information is retained. By doing so, it has effectively minimized the risk of losing sensitive data, bolstering our data security measures. Moreover, the method has resulted in a notable reduction in the number of false positives, which are known to trigger false alarms in intrusion detection systems. Ultimately, this has led to the provision of a high-quality system that meets and exceeds the expectations of our valued customers.

The paper is structured as follows: Section 2 discusses related works, Section 3 presents preliminaries, Section 4 explains the proposed scheme, Section 5 introduces the results and discussion, and Section 6 provides the conclusions.

2. RELATED WORKS

In the fast-evolving world of IoT, researchers are continually developing methods to detect cloud security attacks. ELM remains a key research area due to its efficiency and versatility in handling data. By implementing these approaches, we can effectively detect spammers [1], making it a powerful tool for combating unwanted and malicious activities.

In the study by Dasari et al. [12], a new approach to tackling DDoS attacks is presented, which are known to cause significant disruptions to online services. The method focuses on identifying groups of features that exhibit low correlation, aiming to enhance the detection of UDP-based DDoS attacks using the powerful MLP classification algorithm, the efficient ADAM optimization method and the Tanh activation function.

In the study by Rana et al. [13], this work integrates several machine learning algorithms, such as Support Vector Machines, Naive Bayes, and Random Forests. It was developed on a cloud platform by the "Tor Hammer" attack tool, but this solution did not show much effectiveness.

In their study, Al-A'araji et al. [14] propose an 'Enhanced Intrusion Detection and Classification (EIDC) System' as a firewall to secure the cloud. EIDC detects and classifies received traffic packets using a technique called Nodes 11. Past and current decisions are combined to estimate the final attack category classification.

To enhance security in IoT systems, Hassan et al. [15] propose the use of an artificial neural network (ANN) within both the gateway and application layers of an IoT gateway. The monitoring of subsystem components at the gateway and the overall system state at the application layer is crucial. Once the system was initialized with training data and brought to operating temperature, the researchers intentionally injected inaccurate data into the sensors for 10 minutes. The goal of data execution the neural network will be able to differentiate between valid and invalid data.

In their study, Wani et al. [16] present a system designed to maintain data confidentiality across multiple providers and verify the correctness of user-encrypted data. The one-way proxy (UPRE) to reduce high computational costs with multiple data providers was used. The cloud server embeds noise into the encrypted data, allowing analytics to keep the information confidential.

Chkurbene et al. [17], in their study, proposed a complex method designed to enhance the security of IoT systems. This method provided an early cyberattack detection and response mechanism that was either fully or partially autonomous (i.e., requiring no human interaction). Distributed network security can be ensured with this technique.

Wang et al. [18] suggested a system based on machine learning combinations. To assess accuracy, a fitness function, a genetic algorithm, and a support vector machine (SVM) were integrated. The outcomes demonstrated how well information security is guaranteed by this model.

Samunnisa et al. [19] proposed an anomaly-based intrusion detection system employing hybrid clustering and classification methods to enhance cloud and network security. The main goal is to assess the impact of features on clustering and classification in anomaly-based intrusion detection systems. The study compares and evaluates K-means and other methods in the context of intrusion detection.

In summary, the approaches discussed above focus on the design and development of a security system at the entrance to the cloud and generally rely on the existence of a single centralized cloud. However, there are many requirements to consider, such as resource limitation, distribution, and system scalability. For this, we believe that it is urgent and imperative to develop other approaches that support all aspects of security and present and future attacks.

3. BACKGROUND

3.1 Cloud computing

Cloud computing makes it possible to provide IT services (servers, storage, databases, software, network management, artificial intelligence) [2]. And offer faster and more innovative use, flexible resources and profit at a very high cost and productivity compared to traditional methods. Moreover, there are several types of clouds' which do not necessarily have the same structures and are different in their design and development.

Several types, models and cloud services have been developed to help provide the best solutions for our needs. There are three ways to deploy cloud services for this: public, private or hybrid cloud [5].

Public cloud: is a type of cloud computing in which a service provider makes sharing resources available such as servers and storage to the public via the Internet.

Private cloud: is a type of cloud computing environment dedicated to a single organization. All resources are isolated and under the control of a single unit. This private cloud is also called internal or corporate cloud [19].

Hybrid cloud: Hybrid clouds combine the resources and services of two or more distinct IT environments. Hybrid cloud architectures require integration, orchestration and coordination to be able to move, share and synchronize information quickly [20].

Secure cloud development presents challenges due to emerging consumer security issues. Today, Machine Learning (ML) is a powerful tool used to safeguard the cloud. ML techniques provide a crucial function in preventing and detecting attacks and security breaches in cloud environments [21].

3.2 Overview of ELM

In the realm of cloud-level security, Extreme Learning Machine (ELM) is recognized as an innovative solution. Due to its exceptional efficiency, easy implementation, and ability to handle various tasks like unification, classification, and regression, ELM has become a key area of study. Its potential application in identifying social spammers makes it a compelling choice [20].

In this section, we will briefly discuss the basis of ELM. The ELM algorithm can be summarized in 3 steps [22]:

- Step 1: Definition of hidden layer node number \tilde{N} , randomly assign input weights a_i and hidden layer biases b_i , ($i = 1, 2, \dots, \tilde{N}$).
- Step 2: Calculate the hidden layer output matrix H .
- Step 3: Calculate the output weight β .

The simple ELM learning algorithm has a model of the form: $\hat{Y} = X_2 \sigma(X_1 n)$ where, X_1 is the matrix of input-to-hidden layer weights, σ is an activation function, and X_2 is the matrix of hidden-layer-to-output weights. The algorithm works as follows:

1. Complete X_1 with Gaussian random noise.
2. Estimate X_2 by the least squares method to match the response matrix of the variables Y , use using the pseudo inverse, giving a design matrix T : $X_2 = \sigma(X_1 T) + Y$

This detailed process explains the ELM algorithm: we set N who represent arbitrary distinct samples.

$$(x_i, t_i) \in \mathbb{R}^n \times \mathbb{R}^m$$

where, x_i represents the input sample and t_i define the output sample [23].

The output of a SLFN with L hidden nodes and $g(x)$ which is the function of activation are calculated by the following formulas:

$$O_j = \sum_{i=1}^N \beta_i L_i = 1 (w_i \cdot x_j + b_i) \quad (j = 1, 2, \dots, N)$$

$$\text{Or: } \sum_{i=1}^N \beta_i L(w_i \cdot x_j + b_i) = t_j, \quad j = 1, \dots, N \quad (1)$$

In Figure 1, the computer development of the ELM algorithm is presented in a graphic way.

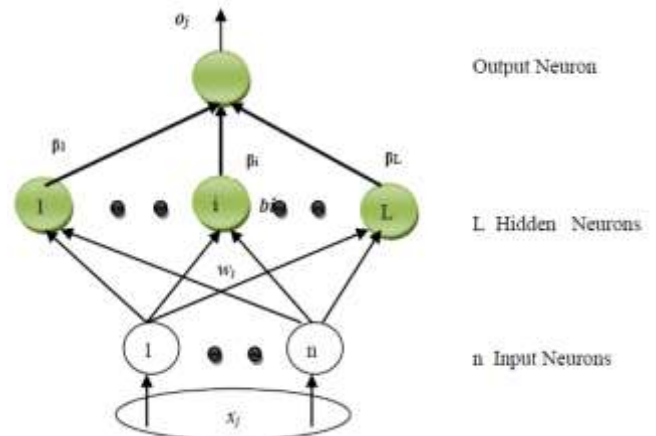


Figure 1. Neurons network

The weights between the input and hidden layer are represented by the $w_i = (w_{i1}, w_{i2}, \dots, w_{in})^T$; these weights are determined through the hidden node i -th. The weights between the two layers (the hidden layer and the output layer) are represented by the b_i ($i = 1, 2, \dots, L$) T . The following formula transforms the non-linear classification into the following linear classification:

$$H\beta = T \tag{2}$$

$H = \{h\}$ ($i = 1 \dots L$ and $j = 1, \dots, N$) is the hidden-layer output matrix.

$h_{ij} = (w_i \cdot x_j + b_i)$ denotes the output of the i^{th} hidden neuron concerning x_j , $T = [t_1, t_2, \dots, t_m]^T$ is the target matrix (classification labels).

Keep in mind that the hidden layer's w_i and b_i node parameters are assigned at random. As a result, the only thing left to figure out for the ELM model is the number of hidden layer nodes, l . The following equation is obtained if the error between the output O_i and the target t can be brought close to zero:

$$\sum_{j=1}^n \|t_j - O_j\| = 0 \tag{3}$$

$$\beta = \begin{pmatrix} \beta^T_1 \\ \vdots \\ \beta^T_L \end{pmatrix}_{L \times m} \quad \text{and} \quad T = \begin{pmatrix} T_1^t \\ \vdots \\ T^t_N \end{pmatrix}_{N \times m} \tag{4}$$

In most studies carried out so far, they have demonstrated that the number of hidden nodes is greatly lower than the number of training samples. Namely ($L \ll N$), with a total of L neurons in the hidden layer [23].

The minimum norm least-square (LS) solution to the linear problem (2) is $\beta = H^+T$, where, H^+ is the Moore-Penrose generalized inverse of matrix H , ELM using such Moore-Penrose (MP) inverse method tends to obtain good generalization performance with highly increased learning speed [24].

4. PROPOSED METHODOLOGY

Considering the different existing methods and approaches and their limitations in the detection of attacks in the cloud, we propose at this work a robust framework to efficiently perform attack detection in the cloud environment. Among the existing security attacks, we are interested in network security attacks. An attacker would want to conduct an indiscriminate integrity assault, for instance, which would result in high rates of false positives and true negatives for classifiers, or they might conduct a targeted privacy violation attack, which would illicitly collect the targeted user's data [25].

Therefore, our main objective is to optimize cloud security in order to reduce the chance of data loss. Finally, if a malevolent user attempts to attack the system, it will be

immediately stopped. In addition, the cloud service provider must only allow authenticated users to access its database. CSP examines the trust values of users to confirm their legitimacy. The user is regarded as authentic if their trust value exceeds the threshold value. Keep in mind that a user's trust value is determined by certain cloud behavior criteria [26]. For use in regression and classification, ELM is a learning technique for single hidden layer feed-forward neural networks. It is more practical than the conventional ANN model and has a simpler and more legitimate mode than the typical BP method. As a result, ELM learns considerably more quickly than BP. ELM tends to attain not only the minimal training error but also a straight solution to the problem.

The proposed method is named attacker detection in Cloud, based on the supervised learning (SL) approach. To filter attackers, all data (text, document, and figure) will be tagged. This process is called document markup.

Figure 2 schematizes our approach. The database will be built from the pre-existing data on the cloud. This data is fragmented into multiple subsets, and then extreme machine learning will be run to make predictions and decisions on each subset of data. Combining the results for each ELM helps distinguish legitimate users from non-legitimate users.

In contrast to alternative approaches like mobile edge computing (MEC) and fog computing (FC) (FC/MEC). Finding the minimal standard of a least squares problem can ultimately be used to convert the straightforward ELM solution into an extended Moore-Penrose inverse problem using a matrix [22].

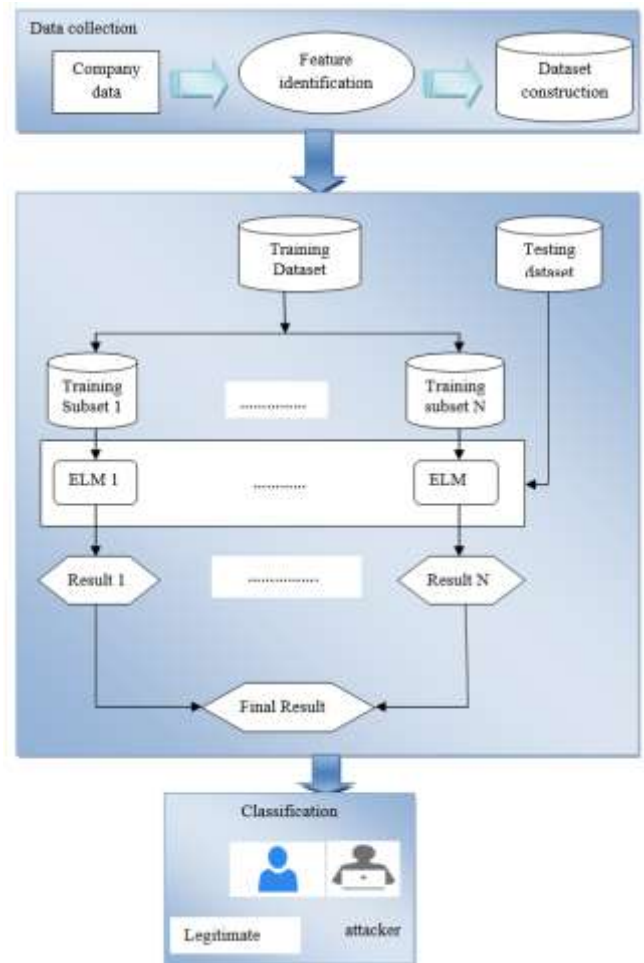


Figure 2. Organization flow of the proposed framework

4.1 Dataset construction in cloud

We will try to gather pre-existing data at the cloud level, which are no classified into non-legitimate users and legitimate users [27, 28]. Unlabeled data collection is the data set containing the most relevant characteristics of multiple cloud user behaviors. However, for the construction of the dataset, the cloud API is used to collect a real dataset from public information. Here we are using K-means clustering which is a type of unsupervised learning used when data is unlabeled [29]. In this step, it suffices to create groups of data represented by the variable K. The algorithm works iteratively to assign each data point to one of the K groups according to the similarity of the characteristics and functionalities provided [24]. Anomaly detection based on user behavior is useful, such as the number of times they log in, the history of these movements and all these activities on the cloud will be evaluated. Next, it is necessary to separate valid and monitored activity groups if a data point moves from one group to another; this should be used to detect significant changes in the data. We summarize our approach as follows:

A first selection of data is created to determine spammers and legitimate users at the level of the cloud network [30]. For that, legitimate users are select from the most active clients in the cloud, for example, users who only work in the cloud.

And non-legitimate users are selected from the set of users who were too often involved in malicious activity example, users who share malicious URLs or messages or, who direct to malicious links, and fictitious websites. Then we generate a list of all users (attacker and legitimate users) by exploring the list of subscribed clients. Liu et al. [31] use a web crawler for this purpose. In addition, each user's behavior is tagged. Then two groups of users were created (Figures 3 and 4) who are, Legitimate and non-legitimate users.

In Figure 5, we show that there is a measurement difference between the original data sent by the legitimate user and the attacker. In general, legitimate users deal with private or public data and share information through the cloud with their friends. But at the same time, most attackers steal and spy on other people's data. Here, we have taken a set of random data, that is stored in the cloud. And taking into account the following parameters: their behavior, the size, the number of executions of this data and the execution time. Thus, two groups of data are formed from its parameters which are legitimate users and malicious users.

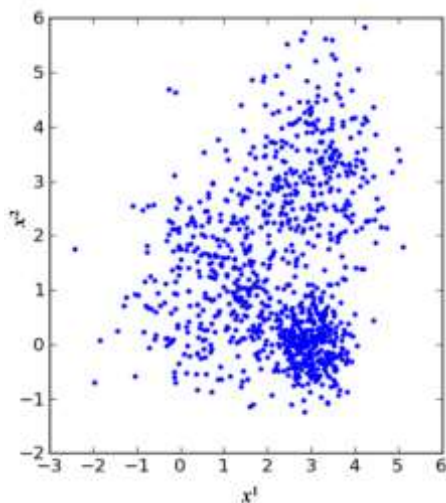


Figure 3. Original unclustered data

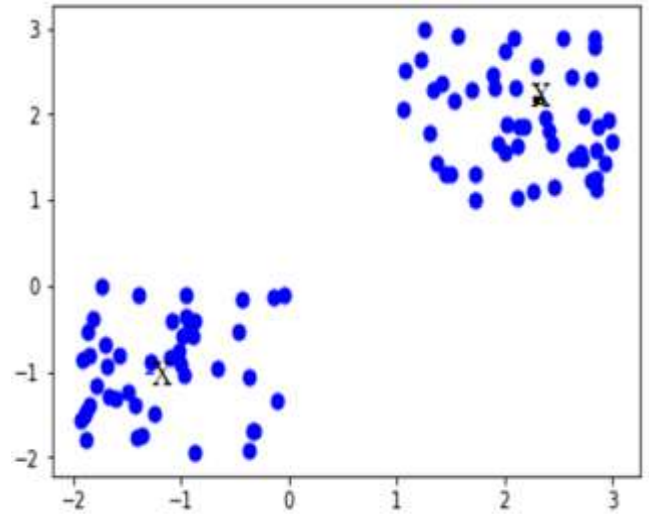


Figure 4. Clustered data

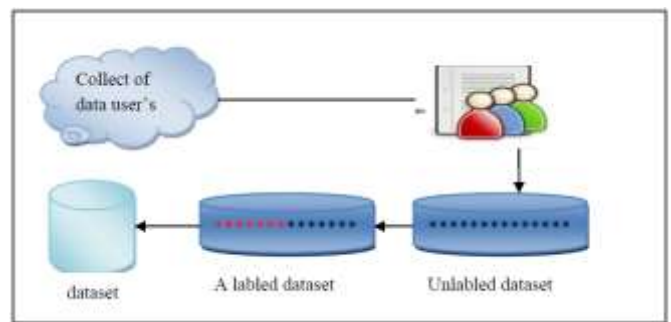


Figure 5. Dataset construction

The process followed by K-Means Clustering (Figure 6) is follows:

- We select the number "K" which determines the number of clusters to deduce.
- Choose a center of gravity to form the cluster. Here we can choose any data point or "k" random points.
- Assign all the data points to their nearest cluster. Let us use the distance method based on the correlation.
- Calculate the centroids of the clusters by taking the average of all the data points that belong to each cluster.

4.2 Training and testing phase

In the test phase for the classification of data and the construction of ELM, initially, we will study a limited amount of data, if the result obtained is satisfactory, we will apply the procedure to a large amount of data by the application of the normal law of probability [32]. Each ELM in our work works as follows:

P: the likelihood of an attack occurring.

q: the inverse of p.

X: represents the number of occurrences per unit of time for an event to occur.

N: the number of experiences.

If P is the probability of an event occurring during a malware detection experiment. And if $q = 1 - p$ is the inverse of P (probability of success), then the probability that this event occurs X times in N experience (i.e. X detection of an attack and N - X no detection) is given by the binomial coefficients [33]:

$$P(X) = \frac{N!}{x!(N-x)!} p^x q^{N-x} = C_N^x p^x q^{N-x}$$

or $X = 0, 1, \dots, N$ and $N! = N \times (N-1) \times (N-2) \times \dots \times 1$.

And 1, $C_N^1, C_N^2, C_N^3 \dots C_N^x$;

But when the number of data becomes very important, it will extend towards the normal law to carry out our test phase. To test our learning machine, we have developed the following test:

We want to determine that our machine was 90% efficient at testing a large amount of data in just 1 minute. Either in a 200-megabyte data sample, we have validated 160 megabytes of correct data, or now we determine if our machine learning is effective. The solution is to let P be the probability of obtaining the correct data; we must then decide on the two following hypotheses (H):

α : low values.

N: amounts of data

q: is the level of significance which is taken at 0.1

H_0 : $P = 0.9$, and our statement is correct.

H_1 : $P < 0.9$, and our statement is false.

We will test for low values of α because we want to know if the proportion of data is too low. If the significance level is taken at 0.1, that is, if the area is grayed, as in the Figure 6, which is equal to 0.1, then $\alpha = -2.33$. The following decision rule is therefore used:

If: H_0 is true, $\mu = NP = 200(0.9) = 180$

And $\delta = \sqrt{NPq} = \sqrt{(200)(0.9)(0.1)} = 4.23$. Then, in reduced centered units: $(160 - 180) / 4.23 = -4.73$. The value significantly lowers than -2.33 , show in Figure 6. Therefore, we conclude that our assertion is justified and that the results are very satisfactory.

K-means cluster Algorithm:

The shaded regions (α) are critical areas, as shown in the graph in Figure 7.

As has been analyzed in several works, the attack can be carried out by introducing erroneous data samples in the training phase to result in a faulty system that does not support any threats [34, 35]. The clarity of the training data and the improvement of the power of the learning algorithms are the two main countermeasures that must be taken into account against any adversary during the training phase (Figure 8).

The data used to train our ML plays a crucial role in the development of a high-level ML model. In general, adversaries prefer to damage training data to minimise the overall performance of the system (Figure 8) [35].

In our framework (Figure 9), the training dataset is divided into K subsets. Each subset contains the same number of samples and p-input features. However, the ELM presents shortcomings for training of big data; like time consumption which is a process of calculating the output matrix of hidden nodes, Moore-Penrose the generalized inverse of a matrix, the Laplace matrix, and matrix multiplications take a long time when forming large-scale datasets.

The testing and training phase determines the reliability of our machine learning. The results of the training phase in Figure 9 is $tr_result = elmk.train(tr_set)$, and the test phase, $te_result = elmk.test(te_set)$ will be combined to obtain at the end, a classification which allows the distinction between legitimate and non-legitimate users by: $print(te_result.get_accuracy)$.

Algorithm

1. // training phase

1- initialize training database with N samples (A_i, Y_i)

2- Randomly initialize W and biases $x_j, j=, 2, \dots, l$

3- Calculate the output weight matrix T_i

4- Calculate $T = H\beta$ where $H_0 =$

$$\begin{pmatrix} L(w_1, x_1, b_1) & \dots & L(w_L, x_L, b_1) \\ L(w_1, x_1, b_N) & \dots & L(w_L, x_L, b_N) \end{pmatrix}$$

$$\beta = \begin{pmatrix} \beta^{T_1} \\ \vdots \\ \beta^{T_L} \end{pmatrix} \text{ and } T = \begin{pmatrix} T_1^t \\ \vdots \\ T_N^t \end{pmatrix}_{N \times m}$$

5- Calculate $\beta = H^* Y_{all}$, where, $Y_{all} = (Y_1, Y_2, \dots, Y_N)$

2. // Detection

a) For each sample i calculate $T_i = \sum_{j=1}^N \beta_j L_i = 1(w_i, x_j, b_i) (j = 1, 2, \dots, N)$

b) Map T_i to Y_i

c) If Y_i represents attack

Then alert

Else

Remain silent

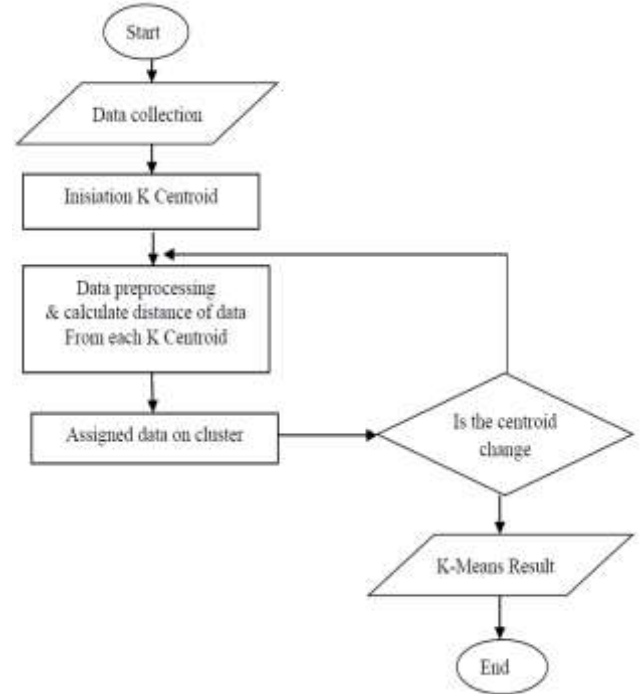


Figure 6. K-means cluster algorithm

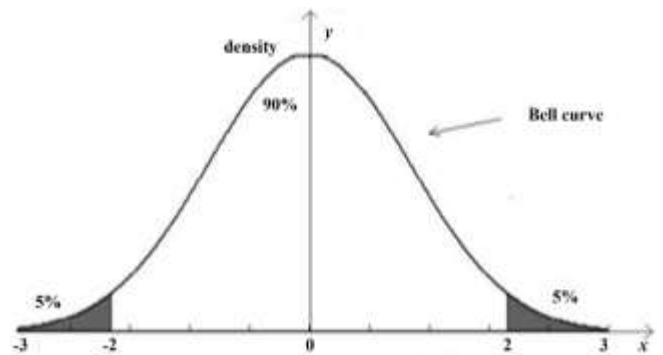


Figure 7. Testing evaluation graph

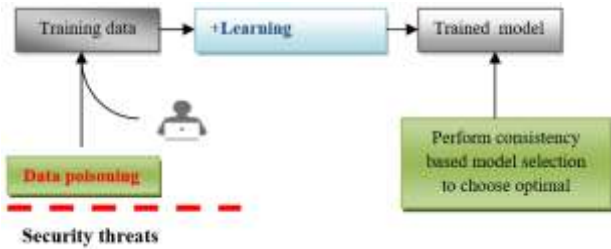


Figure 8. Introducing erroneous data

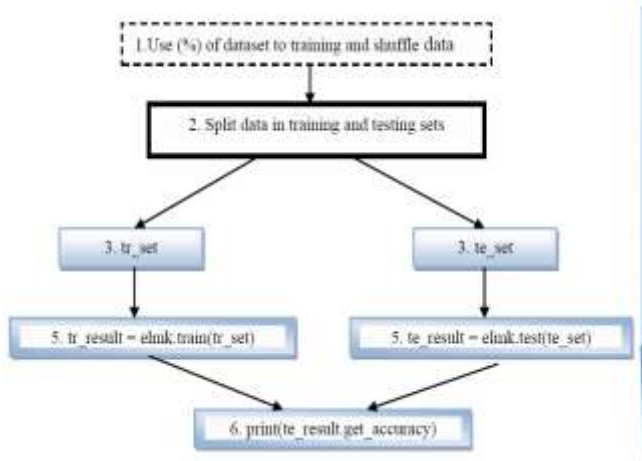


Figure 9. Proposed methodology

When the value is calculated by the equation $\beta = H^* Y_{all}$, that means that, our ELM is trained and ready to detect attacks. It calculates the output for each sample using the equation $T = H \beta$. If the planned release represents an attack, it will generate an alert to the cloud administrator. Otherwise, she remains silent. All data will be tagged to pass the malicious user filtering process. This process is called document labelling.

5. EXPERIMENTAL RESULT

With the rapid development of digital technology, the amount of data circulating on the internet has significantly increased, leading to a corresponding rise in network security threats. Considerably nowadays, it is, therefore, necessary to develop more powerful systems to ensure this security. In this work, we will uncover the remarkable capabilities of our ELM for intrusion detection using the renowned KDD Cup 1999 dataset [36]. Furthermore, we will thoroughly examine the impressive robustness of the State-Preserving Extreme Learning Machine. The assessment of (SPELM) involves utilizing a dimensionality reduction technique such as Principal Component Analysis (PCA) [37]. To gauge the effectiveness of the experimental outcomes, we take into account the following metrics [38]:

- True positive (TP): denotes the number of spammers accurately classified.

- False negative (FN): indicates the number of spammers erroneously categorized as non-spammers.

- False positive (FP): represents the number of non-spammers inaccurately classified as spammers.

- True negative (TN): denotes the number of non-spammers correctly classified.

$$(1) \text{ True positive (TP)} TP = \frac{TP}{TP+FN} \times 100.$$

$$(2) \text{ False negative (FN)} FN = \frac{FN}{FN+TP} \times 100.$$

$$(3) \text{ True negative (TN): } TN = \frac{TN}{TN+FP} \times 100.$$

In our upcoming experiments, we'll be putting ELM, Regularized Extreme Learning Machine (RELM), SPELM, and support vector machine (SVM) to the test in detecting malicious user intrusions on the cloud platform. We'll be using the 1999 KDD Cup dataset for our intrusion detection, and we see how these cutting-edge technologies perform in safeguarding our systems. We conducted all our experiments on a high-performance desktop computer outfitted with an impressive Intel Core i5 Duo CPU E86 @ 3.33 GHz processor and 4 GB of RAM. This powerful setup empowered us to accurately gauge processing time in MATLAB (R2013a) and deliver reliable results.

5.1 Data set description

The classifier learning competition, held alongside the KDD'99 conference, aimed to develop a predictive model (classifier) with the capability to accurately differentiate between legitimate and illegitimate connections within a computer network [34, 39]. The goal was to create a sophisticated system that could effectively identify and classify network activity, contributing to improved cybersecurity measures and network security.

In this experiment, 30,000 normalized and coded digital data samples were used. The results (Table 1) show that our solution correctly identifies 99.2% of non-legitimate users and 99.8% of legitimate users.

Furthermore, we conducted multiple iterations to calculate the mean values for our ELM, RELM, SPELM, and SVM models in terms of training and testing time. The experiences have been carried out several times to have calculated the mean value of each phase. Regarding the values the test and training phase we observe that our model takes a total of 0.0630 seconds for the test and 0.4374 seconds to practice training classification, The detailed experiment results are presented in Table 2. The results indicate a similarity in the performance of the SVM and SPLM models with a slight improvement for RELM.

Table 1. Score of legitimate and nonlegitimate user's

	Legitimate	Non-Legitimate
legitimate	99.8	0.2
non-legitimate	0.8	99.2

Table 2. Comparison between RELM, SPELM, SVM and ELM

Classifier	Training Time (s)	Testing Time (s)
Our ELM	0.4374	0.0630
RELM	0.8091	0.1105
SPELM	1.622	0.0721
SVM	3.031	0.501

Table 3. Test the accuracy of our ELM with RELM and SPELM

# Of Training Samples in %	SPELM (%)	RELM (%)	Proposed ELM (%)
20	97.52	97.81	97.96
30	98.00	97.86	98.10
40	97.89	97.99	98.02

Table 4. Accuracy comparison of proposed ELM with SPELM and RELM using 11 PCs

#Training Samples in %	SPELM (%)	RELM (%)	Our ELM (%)
20	88.17	87.25	91.26
30	88.25	86.36	88.36
40	88.20	85.26	90.28

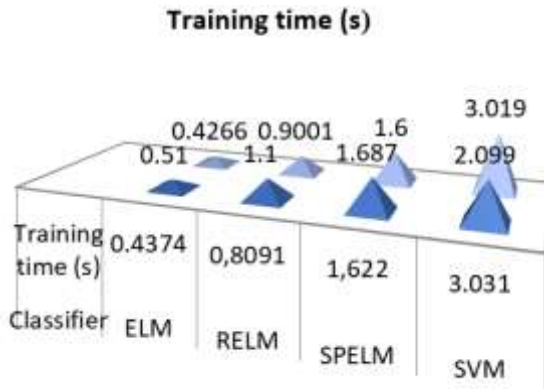


Figure 10. Training time (s) graph

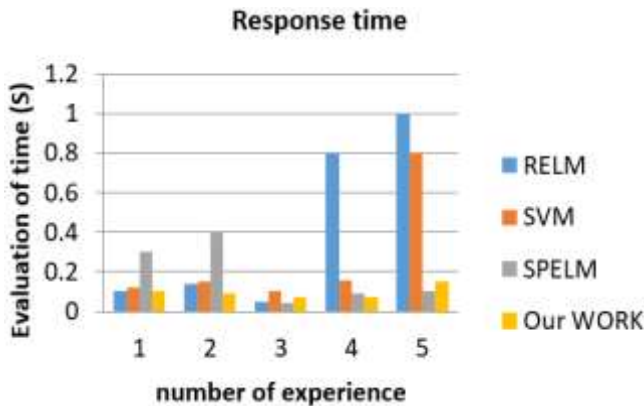


Figure 11. Testing time's graph

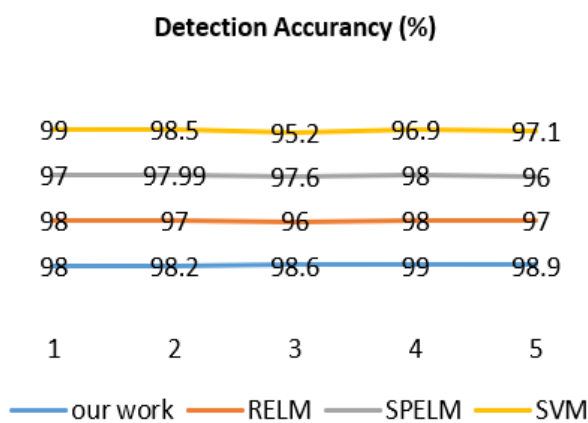


Figure 12. Detection accuracy graph

In our comprehensive evaluation, we meticulously considered the time required for learning from real data and generating synthetic data, along with the rigorous testing process. We then made a comparative analysis of the training and testing duration across our ELM, RELM, SPELM, and SVM. The results portrayed in Table 2 and Figure 10 unmistakably demonstrate the superior speed of our ELM in

comparison to other solutions, establishing its efficiency.

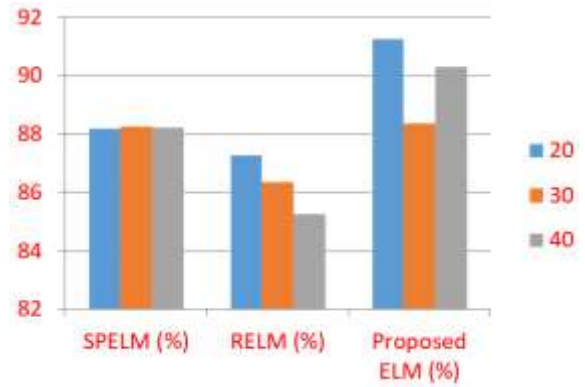


Figure 13. Accuracy comparison graph

In the test phase, we calculate the flow time of the operation compared to the other approaches mentioned above. We have redone the calculation in four (4) different periods. And the results are displayed by the following diagram in Figure 11.

Understanding the performance gaps between different testing methods is crucial to selecting the most appropriate approach based on project-specific time constraints. Substantial variations in training times could significantly affect the practicality and overall effectiveness of the models.

Figure 12 demonstrates the stability of our system compared to others.

During the experimentation, we focused on evaluating parameters related to detection speed and false positive reduction. We closely monitored and recorded significant variations in the results obtained (Table 3).

During the database generation phase, a substantial amount of initial data was eliminated, resulting in a 50% reduction in RAM usage. The tests encompassed a wide array of data, including information from the training phase as well as external data sources, ensuring a comprehensive and thorough analysis. The detailed and comprehensive results, including statistical data and visual representations, can be found in Table 4, while Figure 13 provides a graphical depiction of our findings, offering a nuanced and in-depth view of the results.

5.2 Discussion

In this groundbreaking article, we've unveiled an innovative framework designed to fortify data security in the Cloud. With no labelled data, predefined classes, or centroid, we've harnessed the power of a k-means clustering algorithm to effectively handle untagged data, complemented by the application of an ELM algorithm. By leveraging ELM and the law of least squares, we've successfully tackled the challenge of intrusion detection within cloud networks. Our solution not only boasts reduced training time but also offers exceptional scalability. Furthermore, we've strived to enhance accuracy, surpassing traditional SVM techniques [40, 41]. We have considered a range of solutions and methodologies for establishing a robust intrusion detection system to mitigate the potential loss of control over cloud-based data. If we can detect at least more than 95% of attack connections and filter them out, we can prevent the attacker from overwhelming the cloud server. In the detection of DDoS attacks as show in Figure 14, for example, where attackers install malicious program on the network of vulnerable hosts, and controls managers and robots using a command-and-control mechanism.

In this case, it is necessary to install an attack detection module between the cloud server and the handler, based on ELM. Comparison of the detection accuracy of our work with other proposed works, are shown in the Tables 3 and Table 4 above. Our work is performing well compared to others. However, we need more training time to develop the method of classification as others works.

We also note that false-positive type alerts could generate lot of noise, which sometimes annoys employers in the sector. Let's imagine that more than 200 processes report a false positive alert every 37 seconds. That would require that more than 200 people must sent on-site to detect if there are an anomaly. It seems to us that predicting anomalies using supervised machine learning is the best solution to avoid any potential disaster. And it will be great if we install a system in place to send a signal to the control center in the event of an anomaly. That will help us prevent and stop a disaster problem as quickly as possible before they spread to other linked processes.

Also, in this work, we based ourselves on operational technology (OT) before that of IT technology, with the objective that any application or process developed must be available and used first before being secure. And the machine learning is the best solution in this field. And it cannot in any way be replaced by a human solution.

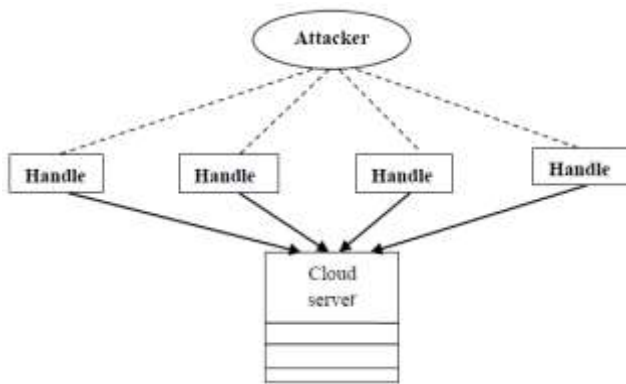


Figure 14. Distributed attack example

6. CONCLUSIONS

In this article, we presented a new approach to bring optimal and reliable security to cloud computing based on ELM techniques. It should be noted that ELM, not only improves characteristics related to classification algorithms but also solves the problem of detecting intrusions in the cloud network reliably and efficiently. And this is what has increased its use in many areas nowadays. The model we introduced is designed to be rapidly developed and tested, making it highly proficient at identifying and detecting attacks with exceptional precision. Its distinct advantage lies in its capability to efficiently classify and organize information. Additionally, the model dynamically and iteratively determines the most suitable number of clusters for grouping, based on pre-existing knowledge.

In summary, this work presents a combined machine learning methodology that addresses the main challenges of information security across the cloud. The results show that the proposed method is effective and efficient. It can be used for larger applications that need real-time performance and

high precision.

In our upcoming work, we are excited to introduce a cutting-edge hybrid clustering approach aimed at revolutionizing the intrusion detection systems performances. By seamlessly integrating multiple clustering methods, including the dynamic K-means and versatile hierarchical clustering, we are ready to shatter the limitations of using a single approach. This innovative approach will involve a meticulous process of analyzing and comparing the results obtained from different clustering techniques, allowing us to harness the strengths of each method while mitigating their weaknesses. Through this methodology, we hope to achieve a comprehensive understanding of how different clustering algorithms can complement each other, ultimately leading to a more robust and effective intrusion detection system. This strategic integration is expected to provide more comprehensive and accurate detection of security breaches, ultimately bolstering our overall security infrastructure.

REFERENCES

- [1] Zheng, X., Zhang, X., Yu, Y., Kechadi, T., Rong, C. (2016). ELM-based spammer detection in social networks. *The Journal of Supercomputing*, 72(8): 2991-3005. <https://doi.org/10.1007/s11227-015-1437-5>
- [2] Rizal, R. (2020). Commonwealth Law Bulletin, Maliciousunauthorised access to computer programs and data in Malaysia. *Commonwealth Law Bulletin*, 47(3): 453-461. <https://doi.org/10.1080/03050718.2020.1835506>
- [3] Anidu, A., Obuzor, Z. (2022). Evaluation of machine learning algorithms on Internet of Things (IoT) malware opcodes. *Handbook of Big Data Analytics and Forensics*, 177-191. https://doi.org/10.1007/978-3-030-74753-4_12
- [4] Guellil, Z., Mahammed, N., Keskes, N. (2024). Distributed k-means clustering using topological relationships. *Ingénierie des Systèmes d'Information*, 29(4): 1297-1304. <https://doi.org/10.18280/isi.290405>
- [5] Soliman, A., Girdzijauskas, S., Bouguelia, M.R., Pashami, S., Nowaczyk, S. (2020). Decentralized and adaptive k-means clustering for non-iid data using hyperloglog counters. In *Advances in Knowledge Discovery and Data Mining: 24th Pacific-Asia Conference*, Singapore, pp. 343-355. https://doi.org/10.1007/978-3-030-47426-3_27
- [6] Subhiyakto, E.R., Rakasiwi, S., Zeniarja, J., Paramita, C., Shidik, G.F., Hasibuan, Z.A., Kesić, M.G. (2024). Evaluation of resampling techniques in CNN-based heartbeat classification. *Ingénierie des Systèmes d'Information*, 29(4): 1323-1332. <https://doi.org/10.18280/isi.2904082015>
- [7] Sun, Z., Wu, Z. (2022). *Handbook of Research on Foundations and Applications of Intelligent Business Analytics*. IGI Global. <https://doi.org/10.4018/978-1-7998-9016-4>
- [8] Guan, Z., Bian, L., Shang, T., Liu, J. (2018). When machine learning meets security issues: A survey. In *2018 IEEE international conference on intelligence and safety for robotics (ISR)*, Shenyang, China, pp. 158-165. <https://doi.org/10.1109/IISR.2018.8535799>
- [9] James, G., Witten, D., Hastie, T., Tibshirani, R. (2021). *Support vector machines*. In *An Introduction to Statistical Learning*. Springer Texts in Statistics, Springer, New

- York, pp. 367-402. https://doi.org/10.1007/978-1-0716-1418-1_9
- [10] Dasari, K.B., Devarakonda, N. (2022). TCP/UDP-based exploitation DDoS attacks detection using AI classification algorithms with common uncorrelated feature subset selected by Pearson, Spearman and Kendall correlation methods. *Revue d'Intelligence Artificielle*, 36(1): 61-71. <https://doi.org/10.18280/ria.360107>
- [11] Chen, W.H., Hsu, S.H., Shen, H.P. (2005). Application of SVM and ANN for intrusion detection. *Computers & Operations Research*, 32(10): 2617-2634. <https://doi.org/10.1016/j.cor.2004.03.019>
- [12] Dasari, K., Mekala, S., Kaka, J.R. (2024). Evaluation of UDP-based DDoS attack detection by neural network classifier with convex optimization and activation functions. *Ingénierie des Systèmes d'Information*, 29(3): 1031-1042. <https://doi.org/10.18280/isi.290321>
- [13] Rana, P., Batra, I., Malik, A., Imoize, A.L., Kim, Y.S., Pani, S.K., Goyal, N., Kumar, A., Rho, S. (2022). Intrusion detection systems in cloud computing paradigm: Analysis and overview. *Complexity*, 2022(1): 3999039. <https://doi.org/10.1155/2022/3999039>
- [14] Al-A'araji, N.H., Al-Mamory, S.O., Al-Shakarchi, A.H. (2021). Classification and clustering based ensemble techniques for intrusion detection systems: A survey. *Journal of Physics: Conference Series*, 1818(1): 012106. <https://doi.org/10.1088/1742-6596/1818/1/012106>
- [15] Hassan, A., Hamza, R., Yan, H., Li, P. (2019). An efficient outsourced privacy preserving machine learning scheme with public verifiability. *IEEE Access*, 7: 146322-146330. <https://doi.org/10.1109/ACCESS.2019.2946202>
- [16] Wani, A.R., Rana, Q.P., Saxena, U., Pandey, N. (2019). Analysis and detection of DDoS attacks on cloud computing environment using machine learning techniques. In 2019 Amity International conference on artificial intelligence (AICAI), Dubai, United Arab Emirates, pp. 870-875. <https://doi.org/10.1109/AICAI45948>
- [17] Chkirbene, Z., Erbad, A., Hamila, R. (2019). A combined decision for secure cloud computing based on machine learning and past information. In 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, pp. 1-6. <https://doi.org/10.1109/WCNC44850>
- [18] Wang, T., Wang, P., Cai, S., Zheng, X., Ma, Y., Jia, W. (2021). Mobile edge-enabled trust evaluation for the Internet of Things. *The Journal of Information Fusion*, 75: 90-100. <https://doi.org/10.1016/j.inffus.2021.04.007>
- [19] Samunnisa, K., Kumar, G.S.V., Madhavi, K. (2023). Intrusion detection system in distributed cloud computing: Hybrid clustering and classification methods. *Measurement: Sensors*, 25: 100612. <https://doi.org/10.1016/j.measen.2022.100612>
- [20] Raza, S., Wallgren, L., Voigt, T. (2013). SVELTE: Real-time intrusion detection in the Internet of Things. *Ad hoc networks*, 11(8): 2661-2674. <https://doi.org/10.1016/j.adhoc.2013.04.014>
- [21] Khilar, P.M., Chaudhari, V., Swain, R.R. (2018). Trust-based access control in cloud computing using machine learning. *Cloud Computing for Geospatial Big Data Analytics: Intelligent Edge, Fog and Mist Computing*, Cham: Springer International Publishing, 49: 55-79. https://doi.org/10.1007/978-3-030-03359-0_3
- [22] Ding, S., Zhao, H., Zhang, Y., Xu, X., Nie, R. (2015). Extreme learning machine: algorithm, theory and applications. *Artificial Intelligence Review*, 44: 103-115. <https://doi.org/10.1007/s10462-013-9405-z>
- [23] Gaurav, D., Tiwari, S.M., Goyal, A., Gandhi, N., Abraham, A. (2020). Machine intelligence-based algorithms for spam filtering on document labeling. *Soft Computing*, 24(13): 9625-9638. <https://doi.org/10.1007/s00500-019-04473-7>
- [24] Zhao, G., Wu, Z., Gao, Y., Niu, G., Wang, Z. L., Zhang, B. (2020). Multi-layer extreme learning machine-based keystroke dynamics identification for intelligent keyboard. *IEEE Sensors Journal*, 21(2): 2324-2333. <https://doi.org/10.1109/JSEN.2020.3019777>
- [25] Zhang, R., Lan, Y., Huang, G.B., Xu, Z.B., Soh, Y.C. (2013). Dynamic extreme learning machine and its approximation capability. *IEEE Transactions on Cybernetics*, 43(6): 2054-2065. <https://doi.org/10.1109/TCYB.2013.2239987>
- [26] Huang, G.B., Wang, D.H., Lan, Y. (2011). Extreme learning machines: A survey. *International Journal of Machine Learning and Cybernetics*, 2: 107-122. <https://doi.org/10.1007/s13042-011-0019-y>
- [27] Koloveas, P., Chantzios, T., Alevizopoulou, S., Skiadopoulos, S., Tryfonopoulos, C. (2021). intime: A machine learning-based framework for gathering and leveraging web data to cyber-threat intelligence. *Electronics*, 10(7): 818. <https://doi.org/10.3390/electronics10070818>
- [28] Muhammad, N.K., Asha, R., Seyit, C. (2021). Lightweight cryptographic protocols for IOT-constrained devices: A survey. *Internet of Things Journal IEEE*, 8(6): 4132-4156. <https://doi.org/10.1109/JIOT.2020.302649>
- [29] Yang, P., Xiong, N., Ren, J. (2020). Data security and privacy protection for cloud storage: A survey. *IEEE Access*, 8: 131723-131740. <https://doi.org/10.1109/ACCESS.2020.3009876>
- [30] O'Connor, C.D., Calkin, D.E., Thompson, M.P. (2017). An empirical machine learning method for predicting potential fire control locations for pre-fire planning and operational fire management. *International Journal of Wildland Fire*, 26(7): 587-597. <https://doi.org/10.1071/WF16135>
- [31] Liu, Q., Li, P., Zhao, W., Cai, W., Yu, S., Leung, V.C. (2018). A survey on security threats and defensive techniques of machine learning: A data driven view. *IEEE Access*, 6: 12103-12117. <https://doi.org/10.1109/ACCESS.2018.2805680>
- [32] Rueda-Rueda, J.S., Portocarrero, J.M. (2021). Framework-based security measures for Internet of Thing: A literature review. *Open Computer Science*, 11(1): 346-354. <https://doi.org/10.1515/comp-2020-0220>
- [33] Aldallal, A., Alisa, F. (2021). Effective intrusion detection system to secure data in cloud using machine learning. *Symmetry*, 13(12): 2306. <https://doi.org/10.3390/sym13122306>
- [34] An, X., Zhou, X., Lü, X., Lin, F., Yang, L. (2018). Sample selected extreme learning machine based intrusion detection in fog computing and MEC. *Wireless Communications and Mobile Computing*, 2018(1): 7472095. <https://doi.org/10.1155/2018/7472095>

- [35] Kanimozhi, A., Vimala, N. (2024). Adaptive Weighted Support Vector Machine classification method for privacy preserving in cloud over big data using hadoop framework. *Multimedia Tools and Applications*, 83: 3879-3893. <https://doi.org/10.1007/s11042-023-15825-9>
- [36] Nassif, A.B., Talib, M.A., Nasir, Q., Albadani, H., Dakalbab, F.M. (2021). Machine learning for cloud security: A systematic review. *IEEE Access*, 9: 20717-20735. <https://doi.org/10.1109/ACCESS.2021.3054129>
- [37] Baraha, S., Biswal, P.K. (2017). Implementation of activation functions for ELM based classifiers. In *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Chennai, India, pp. 1038-1042. <https://doi.org/10.1109/WiSPNET.2017.8299920>
- [38] Charanarur, P., Jain, H., Rao, G.S., Samanta, D., Sengar, S.S., Hewage, C.T. (2023). Machine-Learning-Based Spam Mail Detector. *SN Computer Science*, 4(6): 858. <https://doi.org/10.1007/s42979-023-02330-x>
- [39] Zheng, X., Zhang, X., Yu, Y., Kechadi, T., Rong, C. (2016). ELM-based spammer detection in social networks. *The Journal of Supercomputing*, 72: 2991-3005. <https://doi.org/10.1007/s11227-015-1437-5>
- [40] Elhefnawy, R., Abounaser, H., Badr, A. (2020). A hybrid nested genetic-fuzzy algorithm framework for intrusion detection and attacks. *IEEE Access*, 8: 98218-98233. <https://doi.org/10.1109/ACCESS.2020.2996226>
- [41] Cao, L.L., Huang, W.B., Sun, F.C. (2015). A deep and stable extreme learning approach for classification and regression. In *Proceedings of ELM-2014 Volume 1: Algorithms and Theories*, Springer, Cham, pp. 141-150. https://doi.org/10.1007/978-3-319-14063-6_13