AUIQ Technical Engineering Science

Volume 3 | Issue 1 Article 1

2026

Intrusion Detection in Cloud Computing Using Support Vector Machine and Fast Firefly Algorithm

Ayobami Taiwo Olusesi
Department of Computer Engineering, Olabisi Onabanjo University, Nigeria,
olusesi.ayobami@oouagoiwoye.edu.ng

Ignatius Kema Okakwu
Department of Electrical and Electronics Engineering, Olabisi Onabanjo University, Nigeria.,
okakwu.ignatius@oouagoiwoye.edu.ng

Ayodeji Akinsoji Okubanjo
Department of Electrical and Electronics Engineering, Olabisi Onabanjo University, Nigeria.,
Okubanjo.ayodeji@oouagoiwoye.edu.ng

Ayo Isaac Oyedeji
Department of Computer Engineering, Olabisi Onabanjo University, Nigeria,
ayo.oyedeji@oouagoiwoye.edu.ng

Oluwaseyi Olawale Bello
Department of Computer Engineering, Ekiti State University, Nigeria, bello.oluwaseyi@eksu.edu.ng

Recommended Citation

Olusesi, Ayobami Taiwo; Okakwu, Ignatius Kema; Okubanjo, Ayodeji Akinsoji; Oyedeji, Ayo Isaac; and Bello, Oluwaseyi Olawale (2026) "Intrusion Detection in Cloud Computing Using Support Vector Machine and Fast Firefly Algorithm," *AUIQ Technical Engineering Science*: Vol. 3: Iss. 1, Article 1.
DOI: https://doi.org/10.70645/3078-3437.1052

Scan the QR to view
the full-text article on
the journal website

ORIGINAL STUDY

Intrusion Detection in Cloud Computing Using Support Vector Machine and Fast Firefly Algorithm

Ayobami Taiwo Olusesi [a,*], Ignatius Kema Okakwu [b], Ayodeji Akinsoji Okubanjo [b], Ayo Isaac Oyedeji [a], Oluwaseyi Olawale Bello [c]

[a] Department of Computer Engineering, Olabisi Onabanjo University, Nigeria
[b] Department of Electrical and Electronics Engineering, Olabisi Onabanjo University, Nigeria.
[c] Department of Computer Engineering, Ekiti State University, Nigeria

ABSTRACT

Cloud computing has become popular due to the ongoing developments in the Internet and technical advancements. Users are increasingly storing their Web resources and data in the cloud environment due to the convenience and reduced cost of cloud computing services. Data security is crucial to the development of communication systems in cloud computing. Since data in cloud storage environments needs to be protected from intruders, network security in cloud environment has grown significantly in importance in recent years. In order to address these challenges, there is a need for an intrusion detection system (IDS) that can effectively identify malicious attack in the network. Several approaches have been used in the past years to solve these challenges, but due technological advancement and expansion in cloud users, there is a need to improve on the existing methods. Therefore, this study developed a novel IDS in cloud environment using support vector machine and fast firefly algorithm (SVM-FFA). The developed model was simulated using MATLAB programming environment. Packet delivery ratio (PDR), throughput, energy consumption and accuracy are the performance metrics used in contrasting SVM-FFA with the existing method. In addition to improving user performance in the cloud environment, the study has been able to offer a strong IDS for cloud computing.

Keywords: Cloud computing, Intrusion detection system, Support vector machine, Fast firefly algorithm

1. Introduction

In the quickly changing technology landscape of today, digital transformation has emerged as a key component of organizational success and evolution. More than a trend, this all-encompassing move towards digitalization is a fundamental rethinking of how businesses function, provide value, and engage with their stakeholders via the incorporation of digital technologies [1]. Digital technology rapid development has completely changed how companies run, interact, and handle data. Cloud computing (CC), one of these technological advancements, has become a game-changer for managing data and knowledge. The term “cloud computing” describes a collection of software programs that are made available online as Platform as a Service (PaaS), Infrastructure as a Service (IaaS), or Software as a Service (SaaS) [2, 3]. When combined with a provisioning system that depends on a pay-as-you-go business architecture, cloud computing can lower costs through resource sharing and storage virtualization. The most widely used cloud computing technologies in the software sector are Google App Engine, Simple Storage Service (S3), and Amazon’s Elastic Computing Cloud (EC2) [4]. Economic advantages and cloud infrastructure flexibility have emerged as the main drivers of cloud adoption. Cloud infrastructure has also made processing power and resource scalability available. Consequently, it has made autonomous resource

Received 21 September 2025; revised 28 December 2025; accepted 29 December 2025.
Available online 19 January 2026

* Corresponding author.
E-mail addresses: olusesi.ayobami@oouagoiwoye.edu.ng (A. T. Olusesi), okakwu.ignatius@oouagoiwoye.edu.ng (I. K. Okakwu), okubanjo.ayodeji@oouagoiwoye.edu.ng (A. A. Okubanjo), ayo.oyedeji@oouagoiwoye.edu.ng (A. I. Oyedeji), bello.oluwaseyi@eksu.edu.ng (O. O. Bello).

https://doi.org/10.70645/3078-3437.1052

pooling, on-demand self-service, usage-based charging, resource elasticity, and widespread network access possible [5].

Utilizing cloud computing improves consumer experiences, stimulates innovation, facilitates data-driven decision-making, and has an impact on operational efficiency. Despite the impact and effective services of cloud computing technology, the wide range of uses for cloud computing has made researchers pay more attention to data processing, administration, and storage security [6]. Concerns about the security and privacy of data that is outsourced are raised by cloud computing. In order to accommodate a greater number of users, cloud providers like Google, Microsoft, and Amazon have recently accelerated their cloud computing infrastructure and services. However, as cloud databases typically include sensitive and critical data, the privacy and security concerns will only get worse [5]

Due to the distributed nature of cloud computing, hackers actively target cloud systems in order to take advantage of their weaknesses [7]. Malicious insider attacks have been the most destructive type of attack that has affected all layers of the cloud infrastructure, despite being one of the most ignored. High-level access to network components can grant a malevolent insider root privileges, allowing them to alter private and sensitive information. This assault presents numerous security risks since firewalls and intrusion detection systems avoid detecting such unusual behaviour since they believe it to be legitimate [8]. An intrusion is generally defined as any unauthorized access to the cloud by an intruder, whereas intrusion detection is the process of keeping an eye on and auditing the activities that take place in the computer or network systems. One of the main security concerns of academics is identifying intrusion and stopping network breaches in cloud environments [7]. Therefore, adaptive security measures are necessary for a secure cloud deployment in order to give users a high degree of cloud confidence.

Since the intrusion detection system (IDS) is essential to the security provisioning against intruders, it seeks to maintain the confidentiality, integrity, and availability (CIA) of the networks and information systems in the cloud environment. The three detection approaches in cloud environment are; anomaly-based, signature-based, and hybrid-base. Depending on how the intrusion detection process is carried out [9], IDSs that are anomaly-based find intrusions by searching for unusual patterns in system activity or network events. [10]. The hybrid-based approach offers more comprehensive intrusion detection capabilities by combining the benefits of anomaly-based and signature-based approaches [11]. Statistical analysis, data mining, and machine learning algorithms are some of the techniques used to identify and stop intrusion and detection threats in the cloud. Statistical analysis detection techniques use network computation to identify anomalous behaviours. The benefit of this strategy is that there is no prior knowledge or training on the security threats associated with network traffic. Its inability to identify unusual behaviours because of insufficient knowledge is one of its limitations [12].

The network can recognize a wider variety of unknown attacks if it has access to a large training dataset of attacks. For intrusion detection assaults, cloud computing has embraced machine learning algorithms and has become a widely adopted techniques for intrusion detection in cloud environment [5]. IDS has benefited from the successful application of machine learning (ML) algorithms to increase their efficiency and effectiveness. Machine learning is an appropriate method for network traffic analysis because of its capacity to handle big datasets [13, 14]. In the past years, several machine learning approaches have been used for intrusion detection purpose in cloud environment. To identify network intrusions, suggested a set of classifier models based on tree-based techniques. The system evaluation was conducted using the NSL-KDD dataset. According to the data, employing a sum rule schema to combine classifiers produces far better outcomes than using a single classifier. A network-based intrusion detection system for cloud environments was proposed by [15]. When network traffic deviates from the conventional statistical parameters (such as the average, norm, mode, and standard deviation), it flags the departure. Low-probability occurrences are identified as possible threats by statistics-based intrusion detection systems, which compute a statistical distribution for the typical behaviour profile. [16] take into consideration a machine-learning technique. They suggest a model that makes use of a tiny attack and fault detection system. Learning, trading, and refreshing are the three phases of the protocol's operation. In the trading stage, each hub provides its neighbours with an estimate of its experience. Ultimately, during the refreshing step, trust is updated in accordance with the new standing and the standing is updated based on estimates of expertise. [17] have performed additional work in which they examined the security issues associated with IoT adoption for smart networks. Using a tiered strategy, these writers emphasize various assault techniques. The first layer is known as the "Perception Layer," and it consists of malicious code injection, physical harm, and jamming. Router attacks, floods, spoofing, and traffic analysis are all included in the "Network Layer." Malware, code injection, and social engineering are examples of the "Application Layer" assaults; cryptanalytic, spyware, and distributed denial-of-service (DDoS) attacks are examples of the "Multi-Layer Attacks." [18] examined intrusion detecting service-oriented in vehicular networks. They begin by going over common assaults that are linked to service-oriented networks, such as DoS attacks, Sybil attacks, and attacks that generate false alarms. They assess different intrusion detection systems and suggest a monitoring system that allows cars to turn on an intrusion detection system to keep an eye on their neighbours. They suggest a rule-based intrusion detection method that may be applied to fight against popular attacks like DoS and Sybil attacks in order to lessen their impact.

An overview of several factors to take into account in machine learning intrusion detection systems was given in [19]. An intrusion detection system was presented along with information on classification types, attacks they encountered, and how security incident response infrastructure is set up. A systematic assessment of the literature on anomaly-based intrusion detection techniques, specifically for identifying insider threats, was carried out in [20]. Its goal was to include methods for simulating anomaly detection that is based on both hosts and networks. The study's thoroughness with regard to external assaults or other dangers was constrained by the researcher's exclusive focus on insider attacks. Furthermore, the techniques employed did not align with the most recent advancements and technology in cloud computing. An organized survey of the literature on intrusion detection systems in cloud-based IoT scenarios was provided in [21]. It methodically looked at important papers and crucial methods in this field. The primary problem of IDS, which was detection and precision, was highlighted by the classification of cloud-based IoT IDSs into three basic types: learning-based, pattern-based, and rule-based methods. The study was less thorough for other classes by different criteria because the researcher only divided intrusion detection systems into three categories, which were constrained by the kinds of attacks and contemporary methodologies. Furthermore, the low intrusion detection accuracy suggested that there were no practical ways to address this issue.

The review of related work shows that, IDS in cloud computing still require more attention in the research space. Hence, an anomaly-based IDS for the cloud environment is developed in this study. The developed system combines support vector machine (SVM) algorithm with fast firefly algorithm (FFA). The FFA is an improvement on the standard firefly algorithm (FA). When inbound network traffic flows influence several hosts or virtual machines (VMs) in a cloud setting, IDS can identify potential network assaults. The network administrator is notified to take defensive measures against suspected traffic as soon as any intrusive network events are detected. This paper's remaining sections are organized as follows: The general framework of IDS is presented in section II. The developed IDS is presented in Section III. The results and discussion is covered in section IV. Section V presents the proposed work's conclusion and future directions.

2. The framework of IDS

An observation that substantially departs from established trends and raises the possibility that it was brought about by an unauthorized source or person is called an anomaly. It describes behaviours that compromise network and system security by going against the fundamental ideas of the computer security paradigm, which include confidentiality, integrity, and authentication. Usually, anomalies arise from assaults carried either inside or outside the network, security mechanism breaches, or illegal access. [22, 23].

IDSs are one of the defensive line in recent cybersecurity techniques. It enables organizations to monitor network traffic and system activities for evidence of malicious activity or policy violations, so that the attacks could be detected and reacted to in real-time. The categorization of IDSs is challenging as various systems are tailored to address different operational scenarios, threat models and performance constraints. IDSs can broadly be classified into Network IDS (NIDS), Host-based IDS (HIDS) and hybrid IDS as well as the detection techniques, such as signature based, anomaly based and specification-based systems. In order to detect malicious activity, network intrusion detection systems (NIDSs) continuously examine packets traveling over network links. To offer thorough visibility into traffic flows, these devices are usually installed at key locations in the network infrastructure, such as gateways or between subnets. Fig. 1 shows the framework of the Host-Based IDS which work on personal endpoints, including servers and desktop PCs or virtual machines by inspecting system-level activities. This also involves monitoring logs, process patterns, and filesystem change. An HIDS works well for detecting threats that are aimed directly at the host (e.g., unauthorized entry attempts, malware execution, and privilege elevation). One of the benefits from using a HIDS is its ability to discover local attacks that can bypass network-level observance [24].

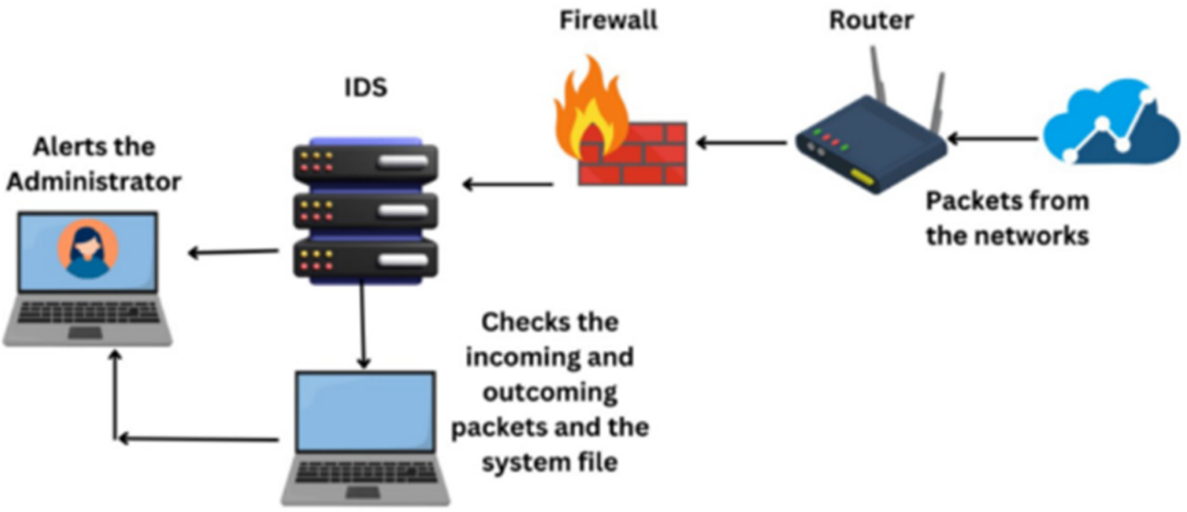
Firewall
Router
IDS
Alerts the Administrator
Packets from the networks
Checks the incoming and outcoming packets and the system file

Fig. 1. Host based intrusion detection system [24].

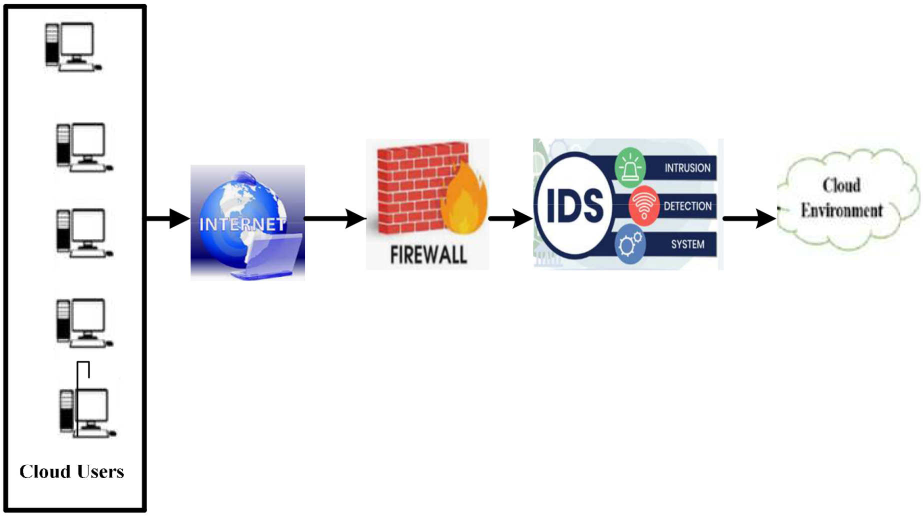

Fig. 2. Intrusion detection system in cloud environment.

3. Materials and method

Presenting an intrusion detection system (IDS) for the cloud environment, as depicted in Fig. 2, is the main goal of this study. IDS is installed at the cloud network entry point and connected to the switches that link the cloud server. Incoming network traffic flows to cloud networks are monitored by the IDS. First, a packet sniffer is used to record the network traffic (raw data). Then, the features of this network data flow are retrieved and well pre-processed before the classification procedure. The most important features are chosen from the incoming network data flow utilizing the FFA in order to increase the intrusion detection process's efficiency and shorten its duration. The anomaly-based detection model receives these chosen features from the SVM-FFA in order to assess and identify their nature. When suspected traffic is identified, the cloud network administrator is alerted to block the source IPs of the suspected traffic and drop it, while permitting normal traffic to pass through. The network data source for the developed system is the benchmark NSL-KDD dataset.

3.1. Data set

A standardized dataset is necessary in order to assess any IDS's performance. Several datasets such as DARPA-98 dataset, KDD Cup 99 among others have been extensively used to evaluate IDS models during the past years. Nevertheless, a number of problems with the dataset were found, such as the inclusion of synthetic traffic, imbalance in test data, and duplicate

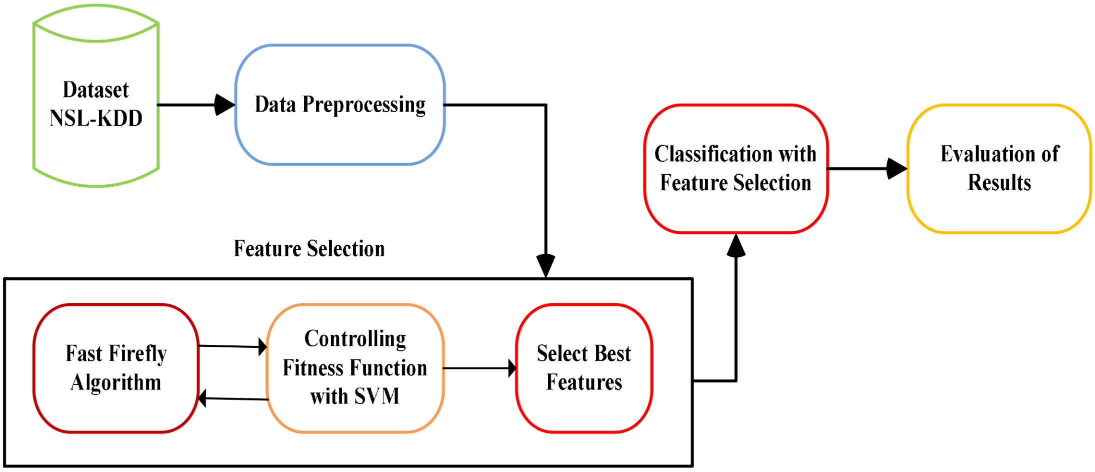

Fig. 3. Flow chart of SVM-FFA model.

records. Also, a number of important problems that impair system performance and cause anomaly detection techniques to produce subpar predictions. Research has been conducted to solve these constraints by developing improved datasets, including the NSL-KDD dataset. Records from the KDD dataset that have been carefully chosen are included in this dataset. Hence this study utilized the NSL-KDD data set for proper and accurate detections of intrusions in cloud environment. The NSL-KDD dataset has the significant benefit of removing duplicate and redundant records from both the training and test datasets, which lowers the quantity of false positives in classification results [24].

3.1.1. Data Pre-processing

Large amounts of data about network traffic comprise a wide variety of attributes with different types and ranges of values. As a result, processing these data directly entails laborious procedures that lead to incorrect classifications, which are unacceptable in IDSs. The following is a breakdown of the data preparation techniques:

3.1.2. Data mapping

Nominal forms are used to represent several aspects of the network relationships. These symbolic data are too complex for the majority of classifiers to handle directly. As a result, these characteristics have to be converted to a numerical format. The feature value of the class label is changed to either 0 for the normal classes or 1 for the other abnormal classes.

3.1.3. Data normalization

There is a bias toward some network attributes over others since network connections contain a wide variety of features with different values. The bias issue can be avoided by normalizing these characteristics. By transforming the connection instances to a standard form and scaling their feature values to a predetermined range based on a certain normalization method, this improves the effectiveness of the IDS. Using Eq. (1), the Z-Score normalization technique scales the feature values of each network connection X [25].

$$\text{Data Normalization}\,(X_{id}) = \frac{X_{id} - mean\ \left(feature_d\right)}{std\ \left(feature_d\right)} \tag{1}$$

i is the instance number of connection and d is the dimension of the feature network.

3.2. Feature selection

As shown in Fig. 3., the feature selection stage, which comes after data preprocessing, is an essential part of this system. Feature selection is essential for boosting classification models' performance since it removes redundant or irrelevant features while increasing accuracy and efficiency [24]. It is evident that irrelevant features cause the training and testing procedures to lag, the classification accuracy to drop, and the detection time of the employed detection model to increase, all of which are unsuitable for IDS [26, 27]. In fact, removing the superfluous features speeds up calculations, reduces the complexity of the detection model, overcomes model overfitting, and creates a lightweight IDS. By choosing the more relevant and ideal features subset from the incoming network data flow, meta-heuristic techniques are used to optimize performance evaluation by methodically choosing and reducing features inside datasets.

In this study, the FFA which is one of the meta-heuristic techniques was used in order to choose, minimize the features of datasets and to reduce the dimensionality of the network connection. A binary vector of length N, where N is the number of network connection attributes, is displayed for each potential solution. If the feature value is set to 0, it will be removed from the feature subset, and if it is set to 1, it will be retained. The SVM detection accuracy, which serves as the FFA fitness function, is utilized to assess each feature subset.

From the optimization perspective, feature selection is combinatorial task. In a data set with h features, the total number of possible subset is 2^h, which in turn becomes computationally prohibitive for large data set h. Therefore, the fast firefly fly algorithm (FFA) which is the meta heuristic algorithm of this study was used to efficiently search for optimal subsets. The focus of feature selection is made up of two folds. The first one is to maximise classification accuracy using the selected feature subset while the second aspect is to, minimize the number of selected features so as to reduce the complexity.

To achieve this, this study defines a scalarized objective function as shown in Eq. (2) in order to balance the two focuses of the feature selection.

Objective function:

$$(F) = g.\left(1 - \frac{|F|}{h}\right) + (1 - g)\,.Acc\,(F) \tag{2}$$

F is the selected feature subset, $|F|$ is the number of features in F, h is the total number of features, and $Acc(F)$ is the classification accuracy using subset F. The parameter $g \in [0, 1]$ controls the trade-off between minimizing feature count and maximizing accuracy. A higher g provides feature reduction while a lower g provides accuracy.

3.3. Data classification

In this study, a classification model is applied for the detection of network intrusion in the NSL-KDD dataset using Support Vector Machine (SVM). The SVM is faced with the challenge of optimizing its control parameters which are gamma and kernel coefficient (C). Hence this study made used of grid search approach for to optimize SVM parameters. The SCVM with grid search detail is discussed below:

3.3.1. Support vector machine

A supervised learning method for classification training in a variety of domains is the Support Vector Machine (SVM). The SVM model is appropriate for both linear and non-linear data classification applications, and it may be used for binary and multi-class classification. In high-dimensional space, the SVM model generates a hyper-plane for data partitioning and chooses the optimal hyper-plane based on its ability to divide the data. Different kernel functions are used to estimate the margins of the non-linear classifier; the radial basis, linear, polynomial, and sigmoid kernel functions are frequently used. Because of its effectiveness in classification, the SVM model has been effectively used by researchers in a variety of applications, including pattern recognition and image processing [28]. In this study, the SVM with the radial basis function (RBF) is used as the detection model as shown in Eq. (3).

$$K\left(x, y\right) = e^{-\frac{\left|x_i - x_j\right|^2}{2\Upsilon^2}},\ \Upsilon > 0 \tag{3}$$

For training samples $(x_i,\ y_i),\ i = 1, 2, \ldots .n$ where the maximum number of samples is given as $i, y_i \epsilon\, [1, -1]$. The technique of generating optimal is carried out in Eqs. (4) and (5).

$$min \frac{1}{2} ||\omega||^2 \tag{4}$$

$$y_i\left(wx_i + b\right) \geq 1,\ i = 1, \ldots \eta \tag{5}$$

The above Eq. (5) utilized increase of $||\omega||^2$ value along with concentration with $y_i(wx_i + b)$. If the result of the information is $y_i = +1$, then $y_i(wx_i + b)$ becomes $(wx_i + b) \geq +1$ as seen in Eq. (6). If the result of the information is $y_i = -1$, then $y_i(wx_i + b)$ becomes $(wx_i + b) \geq -1$ as seen in Eq. (7). If the model is unable to identify a hyper-plane, soft margin is applied. As seen in Eqs. (6) to (8), the positive slack variables in soft margin use $\eta_i,\ i = 1,\ 2,\ \ldots ..N$ in the constraints.

$$\left(wx_i + b\right) \geq +1 - \eta_i \text{ for } y_i = +1 \tag{6}$$

$$\left(wx_i - b\right) \geq -1 + \eta_i \text{ for } y_i = -1 \tag{7}$$

$$\eta \geq 0 \tag{8}$$

In the event of an error, η_i must be greater than unity. The training error then has an upper constraint of $\sum_i \mu_i$. Eq. (9) provides the Lagrange.

$$Lp = \frac{1}{2}/w^2/ + C\sum_{i=1}^{n} \eta_i - \sum_i \alpha_i \left\{y_i\left(x_i.w - b\right) - 1 + \eta_i\right\} - \sum_i \beta_i \mu_i \tag{9}$$

where Lagrange multipliers are employed to produce μ_i positive values and are represented as β_i. Thus, the malicious attackers' information is provided by the SVM and disseminated around the network. As a result, the neighbouring user's ID gets deleted from malevolent attackers. The SVM's performance is greatly influenced by the values of its control parameters. Therefore, choosing its parameters is crucial. These include the RBF kernel parameter (σ), which controls the correlation between the support vectors and establishes the nonlinear mapping to a higher dimensional space, and the penalty parameter (C), which regulates the flexibility of the separable hyperplane. SVM model has the advantage of requiring few parameters, but it has the drawback of requiring a Gaussian function in the training set for every instance, which degrades performance and lengthens training times on big datasets in classification cases. The grid search approach is the searching technique to optimize the control parameters [C, σ] of the SVM in order to maximize its performance. The grid search improves the IDS performance by reducing the noise and the false positives of the IDS.

3.3.2. Grid search

To improve the classification accuracy of SVM, its enhanced by optimally using grid search. The grid search is the selection of integration methods and hyperparameters through testing and validating every integration. Fig. 5 illustrate the process of optimally selecting the parameters using grid search optimisation method. The aim of grid search is to regulate integration that generates the best performance for model prediction. Grid search in integrated with k-fold cross validation (CV) which is an estimation index for SVM classification. The k-fold CV is repeatedly testing the data many times till the k repetitions utilized the data as test information. The accuracy of k methods is acquired and the performance of k-fold cv classification method is estimated. The classification parameters (gamma and C) are optimally modified by grid search. The optimization procedure is shown in Fig. 4, where i is the current iteration and n is the maximum number of iterations. The SVM is tuned by the grid search to generate the highest accuracy score thereby reducing the noise and false positives of the IDS.

3.4. Firefly algorithm

The Firefly Algorithm draws inspiration from the way fireflies naturally approach one another in the dark by utilizing their self-luminosity. The behaviour of fireflies is based on three assumptions. Firstly, all fireflies are unisex and as a result, either male or female gender can be drawn to one another. Second,

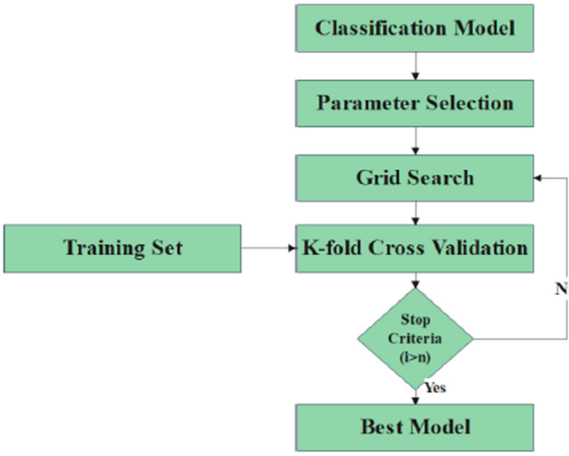

Fig. 4. Grid search optimization process.

the intensity, which depends on the distance between the firefly in question and the others, is related to the attractiveness. As the distance grows, the appeal diminishes. Lastly, cost function value provides luminous intensity of a firefly [29]. Mathematical expression of the FA algorithm is shown from Eqs. (10) to (13).

Eq. (8) provides the firefly's light intensity.

$$I\ (r) = I_0 \exp\mu\left(-\mu r_{ij}\right) \tag{10}$$

Where μ is coefficient of absorption and value of initial is I_0 at $r = 0$

Eq. (10) expressed the attractiveness where β_0 is the initial value at ($r = 0$)

$$\beta = \ \beta_0 \exp\left(-\mu . r_{ij}^{m}\right),\ m \geq 1 \tag{11}$$

Eq. (11) calculates the cartesian distance between two fireflies, i and j, at coordinates x_i and x_j, respectively. where the dimensionality of the problem is indicated by D and x_{ik} at the k_{th} element of the spatial coordinate x_j of the i_{th} firefly.

$$r_{ij} = /r_i - r_j/\ = \sqrt{\sum_{k=1}^{D}\left(x_{ik} - x_{jk}\right)^2} \tag{12}$$

Equation motion of the i^{th} to the j^{th} one is determined by Eq. (13)

$$x_i\left(t+1\right) = x_i\left(t\right) + \beta\left(x_j\left(t\right) - x_i\left(t\right)\right) + \alpha\left(rand - 0.5\right) \tag{13}$$

$x_i(t+1)$ is the firefly position i at iteration $t+1$. Equation (4) is made up of three parts. The first part $(x_i(t))$ is the firefly position i at iteration t, $\beta(x_j(t) - x_i(t))$ is relative to the attractiveness forms

Table 1. Firefly algorithm [29].

```
// Parameters initialization of FA (Population size, $\alpha$, $\beta_0$, $\mu$ and number of iterations)
The Light intensity is defined by the cost function $f(x_i)$ where $x_i$ $(i = 1, \ldots\ldots n)$
While (iter< Max Genertaion)
    for i = 1: n (all n fireflies)
        for j = 1:n (all n fireflies)
            if $(f(x_i) < f(x_j))$, where firefly i towards j,
            end if.
            Update attractiveness $\beta$ with distance r.
            Evaluate new solution and update $f(x_i)$ in the same way as Eq. (13)
    end for j
    end for i
rank the solutions and find the best global optimal.
end while
Generate the output
```

the second part and the third part ($\alpha(rand - 0.5)$) is randomization (flying blindly with no light) where α is the parameter of random walk $\alpha \epsilon [0, 1]$ The firefly algorithm is shown in Table 1

3.5. Fast firefly algorithm (FFA)

It is important to note that the original approach of FA runs (Max generation n.n) tests. However, in the adopted FFA only (K.n) tests are done, where K is an integer. It indicates that FFA takes a lot less time than the traditional one. The new position of Eq. (10) is modified to Eq. (14):

$$x_i\,(t+1) = \alpha.x_i\,(t) \tag{14}$$

In the original version of FA, the values of α and μ are empirically determined based on each test function, with β_0 equal to unity. On the other hand, the α in FFA is regarded as equivalent to:

$$\alpha = \exp\mu\,(-10.iter/\,(iter + 10)) \tag{15}$$

where α is still selected to equal 1 and convergence is easily attained. Rather of being constant, the randomization parameter α is exponentially reduced. The research balance between exploitation and exploration of the improved FFA can yield greater results than its rival FA using this inject artifice. It would be possible to speed up the process of updating the firefly motion in the original FA version. To easily reach the overall optimum, each firefly in the swarm can benefit by reorienting its motion to select a promising spot. In order to improve the algorithm's exploration and exploitation, the updated term is redirected, and the speed of its convergence is thus assured [29, 30]. The main idea behind the adopted FFA is to improve the capacity of FA technique while maintaining a reasonable search efficiency. It indicates that the optimal solution for the numerous benchmark functions and other applications was obtained using (K.n) assessed tests, which were found to be obviously sufficient. The SVM-FFA algorithm is shown in Table 2.

3.6. Intrusion detection based on SVM-FFA

This section gave the illustration of the developed FFA with SVM to enhance the detection accuracy of intrusion in cloud environment. Fig. 5 shows how

Table 2. SVM-Fast firefly algorithm.

```
// Parameters initialization of FA (Population size, $\alpha$, $\beta_0$, $\mu$ and number of iterations)
The Light intensity is defined by the cost function $f(x_i)$ where $x_i$ $(i = 1, \ldots\ldots N)$
        While (iter< Max Genertaion) do
                for k = 1:K.n (all n fireflies) // this is the first modification
                i = rand(n) // this is the second modification
              j = rand(n) // this is the third modification
                Calculate its fitness using SVM classifier
                if $(f(x_i) < f(x_j))$, move firefly i towards j.
                end if.
                Update attractiveness $\beta$ with distance r.
                Evaluate new solution and update $f(x_i)$ in the same way as Eq. (11)
                Modify the new position obtained by Eq. (11) according to Equation (12)
                end for k
    rank the solutions and find the best global optimal
    end while
        Generate the output
```

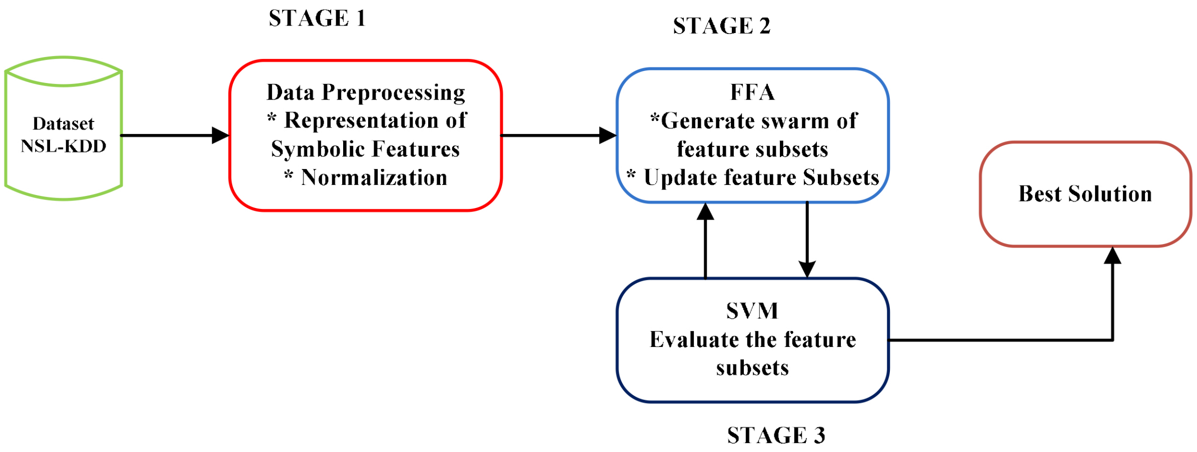

Fig. 5. Integration of SVM with FFA.

FFA was integrated with SVM for feature selection of relevant dataset. The NSL-KDD data set was used to evaluate the performance of the model. The SVM-FFA is made up of three stages. In the first stage, data preprocessing is achieved where the symbolic features are converted to numeric ones. Thereafter, the data is set to [0, 1]. Then, in the second stage the FFA is applied for the purpose of building a swarm of subsets and optimal feature selection of dataset. In the third stage, optimized feature subsets were passed into the SVM for classification and evaluation. The second and third stages are repeated so as to reach the best and optimal subset of features as shown in Fig. 5.

3.7. Parameters of SVM-FFA algorithm

As shown in Table 3, the number of populations size is set as 50 and the number of iterations is set at 50In the process of transforming the data space entries in a space of large dimension called the feature space. There is a possibility of dividing line in the feature space, hence in cases where the dataset is not linearly separated, the kernel function of SVM help to deal with such situation. Few of the types of kernel functions are the linear kernel, the polynomial kernel, radial basis function (RBF) kernel and sigmoid kernel. In this study, the RBF was chosen as the kernel function. The RBF is made up of two major parameters which are the gamma (g) and regularization parameter (C). These two parameters were optimized using the grid search approach Studies have shown that, (g) and (C) has a great impact in the performance of SVM.

Table 3. SVM-FFA parameters.

Parameter	Values
Population Size	50
Number of folds in cross-validation	10
Attrractiveness (β_0)	1.0
Kernel Type	RBF
Kernel Regularization (C)	[0.1,1,10]
Kernel Coefficient (Gamma)	[0.01,0.1,1]
Absorption Coefficient (γ)	1.0
Number of Iteration	50

4. Results and discussion

The SVM-FFA technique is used in cloud computing and assessed a number of metrics. The experiment and simulation of this study was implemented on Intel Core i7 CPU, 8GB of RAM, 500GB Solid State Drive (SSD), MATLAB programming environment, and Windows 10 were among the tools and software utilized. MATLAB is a computer programming tool that offers an appropriate environment to the user and high performance for technical computing which allows immediate testing of an algorithm without recompilation. It also facilitates the algorithm implementation process which requires both iterations and precise functions in getting optimal solutions. It also contains libraries that provide symbolic functions and graphic plots of the obtained results. Table 4 shows the simulation parameters.

Table 4. Simulation parameters.

Parameter	Value
Number of Users	50–500
Network Area	$1000m \times 1000m$
Propagation Model	Propagation/TwoRayGround
Mac type	Mac/802_11
Classification model	SVM
Feature selection	FFA
Channel type	Channel/Wireless Channel
Phy type	Phy/WirelessPhy
Packet Size	20 bits
Antenna type	Omni Antenna

To assess the model's effectiveness, the SVM-FFA technique is contrasted with the existing approach of the traditional firefly algorithm (FA) and weighted coefficient firefly optimisation algorithm (WCFOA). Throughput, energy consumption, accuracy, and packet delivery ratio are the performance metrics which are defined and expressed mathematically below.

4.1. Packet deliver ratio (PDR)

This is the total number of received data packets divided by the number of data packets sent is known as the packet delivery ratio. Eq. (16) is used to determine the data PDR.

$$\text{PDR} = \left(\frac{\sum corectly\ recived\ data}{\sum sent\ data\ packets}\right) \times 100 \tag{16}$$

4.2. Energy consumption (EC)

It is measured as the quantity of energy used by users for tasks like data forwarding and sensing. The unit of measurement for this is joules and the mathematical expression is shown in Eq. (17).

$$\text{EC} = \sum_{i=1}^{n} Sn_i \times EC\,(SSn) \tag{17}$$

Sn_i is the individual user i in the network, n is the total number of users, $EC(SSn)$ is the energy consumption for single user Sn_i

4.3. Accuracy (A)

This is defined as the ratio of correctly classified samples to all samples. Equation Eq. (18) provides the method for calculating detection accuracy.

$$\text{A} = \left(\frac{TP + TN}{TP + TN + FP + FN}\right) \times 100 \tag{18}$$

TP is true positive, TN is true negative, FP is false positive and FN is false negative

4.4. Throughput

It describes the maximum size of data packets that can be properly transmitted in a specific amount of time. Eq. (19) shows how throughput is measured.

$$\text{T} = \frac{SPT \times P_s}{time\ \ (\text{sec})} \tag{19}$$

T is the throughput, SPT is the size of successfully delivered data packet, P_s is the size of packet.

As shown in Fig. 6 and Table 5, the SVM-FFA model packet delivery ratio is calculated and compared with existing technique of FA and WCFOA. In order to optimally select the relevant features, balance exploration, exploitation throughout the search phase, the developed IDS model adopted the FFA technique. Exploration and exploitation of SVM- FFA were able to balanced the network, detect attacks, minimize packet loss and improve the packet delivery. The PDR of the SVM-FFA model is higher than that of the existing cloud computing techniques. The overfitting issues in the model and the local optima trap are limitations of the existing approaches. For 450 users, SVM-FFA technique has 98% PDR while FA and WCFOA technique has 87 and 94% PDR respectively. The network's correctly transmitted packets at a particular time will undoubtedly improve when a network intrusion is detected at the right moment before deeply infiltrating the network. And Fig. 7 and Table 6 illustrate what the developed SVM-FFA was able to accomplish. The SVM-FFA model throughput was compared with the existing FA and

Table 5. Packet delivery ratio of SVM-FFA compared with FA and WCFOA.

Users	FA	WCFOA	SVM-FFA
0	0	0	0
50	80	91	92
100	82	96	97
150	85	94	96
200	84	92	98
250	86	95	97
300	83	93	95
350	84	95	93
400	86	94	98
450	87	94	98
500	85	96	97

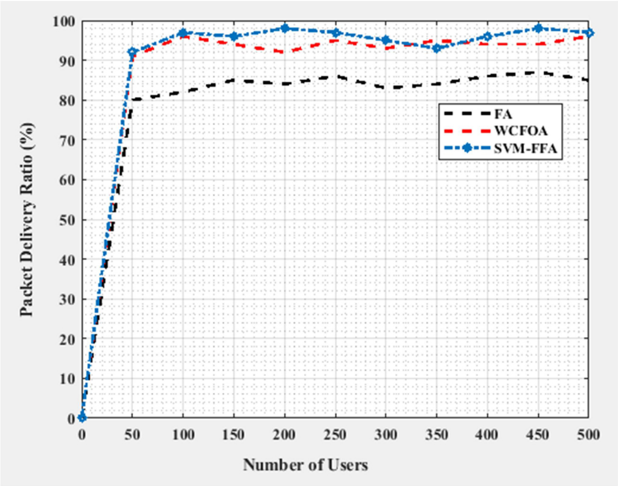

Fig. 6. Packet delivery ratio.

Table 6. Throughput of SVM-FFA compared with FA and WCFOA.

Users	FA	WCFOA	SVM-FFA
0	0	0	0
50	9	14	20
100	12	15	25
150	14	17	28
200	16	19	34
250	19	20	36
300	20	22	39
350	23	29	43
400	25	29	43
450	29	35	48
500	31	35	52

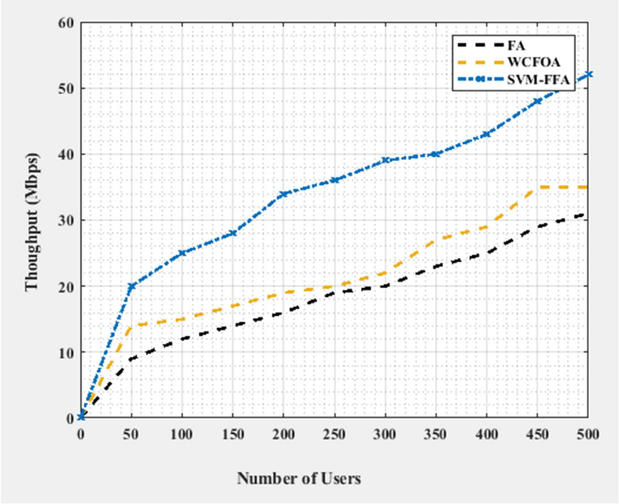

Fig. 7. Throughput.

Table 7. Energy consumption of SVM-FFA compared with FA and WCFOA.

Users	FA	WCFOA	SVM-FFA
0	0	0	0
50	20	21	10
100	24	21	12
150	27	22	13
200	29	23	15
250	31	23	17
300	33	24	18
350	36	25	20
400	38	25	21
450	39	27	22
500	41	27	24

WCFOA techniques The developed SVM-FFA model has a greater throughput than FA and WCFOA. The model's overfitting problems and the local optima trap are drawbacks of the current methods. The SVM-FFA technique has a 39Mbps for 300 users, whereas the FA and WCFOA techniques have corresponding throughput of 20 and 22Mbps respectively. Energy consumption by the users within the network is also one of the key performance metrics that determine the how a long a network will be able to stay and also improve the network lifespan. As shown in Fig. 8 and Table 7, the SVM-FFA model's energy usage is

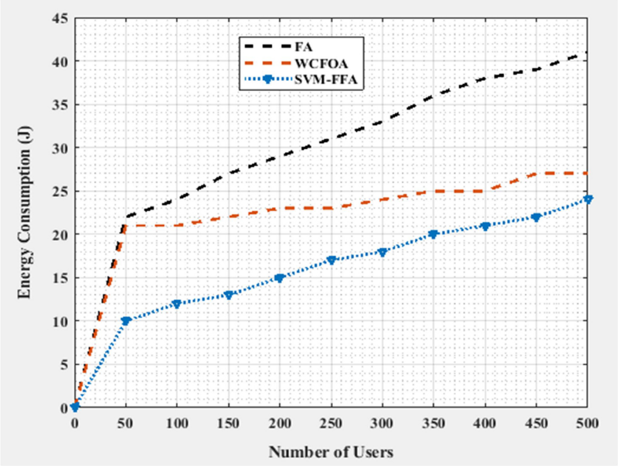

Fig. 8. Energy consumption.

Table 8. Accuracy of SVM-FFA compared with FA and WCFOA.

Users	FA	WCFOA	SVM-FFA
0	0	0	0
50	78	90	92
100	79	90	93
150	81	90	93
200	82	91	95
250	84	93	95
300	85	94	96
350	86	96	97
400	88	96	98
450	90	96	98
500	92	96	98

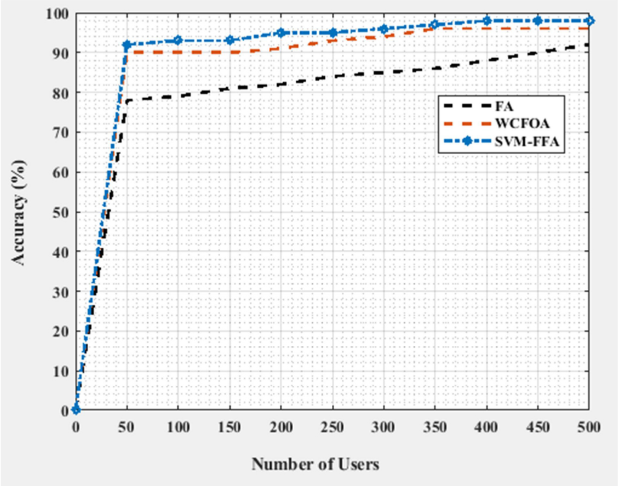

Fig. 9. Accuracy.

measured for different users and contrasted with current methods. The SVM-FFA enables in striking a balance between exploitation and exploration. This lowers the amount of energy used for exploration or exploitation. The SVM-FFA approach uses less energy than other approaches. For 450 users, SVM-FFA uses 22 J while FA and WCFOA use 39 and 27 J of energy respectively. In terms of accuracy for intrusion detection in cloud environment using SVM-FFA, as shown in Fig. 9 and Table 8, the developed SVM-FFA has a better accuracy performance than the existing FA and WCFOA. For 500 users, SVM-FFA has a detection accuracy rate of 98% while FA and WCFOA has a detection accuracy rate of 92 and 96% respectively.

5. Conclusion

Cloud computing is a distributed system, which has made it possible for intruders to aggressively target cloud systems and exploiting their vulnerabilities. This has become a primary security concern for researchers in cloud systems. Therefore, to give cloud users a high level of confidence, there is need for a secure cloud deployment. This study developed an intrusion detection system (IDS) in cloud computing by combining the mechanism of support vector machines (SVM) and the fast firefly algorithm (FFA). The SVM model can be applied to binary and multi-class classification and is suitable for both linear and non-linear data classification applications. The SVM model creates a hyper-plane for data partitioning in high-dimensional space (such as cloud environment) and selects the best feature depending on how well it divides the data. This study uses FFA in feature selection to enhance the performance of the classification (SVM) models because it eliminates superfluous or irrelevant features while boosting efficiency and accuracy. The SVM-FFA technique shows a better performance in terms of energy consumption, throughput, packet delivery ratio and accuracy in contrast to the existing method of intrusion detection. This performance of SVM-FFA has help to improve the network lifespan. Because a secured network with increase in its PDR, accuracy, throughput and less energy consumption will definitely have a prolong performance. The future work of this study can look into comparing SVM-FFA with some of the deep learning based IDS technique such as the convolutional neural network (CNN), long short-term memory (LSTM), recurrent neural network (RNN) among others.

Acknowledgment

The principal author would like to thank the other authors for their efforts and supports. We would also like to thank the reviewers for their contributions in improving our work.

Conflict of interest

Regarding this study, the authors declare that there are no conflicts of interest.

Funding

No funding was received for this research

References

1. L.J. Victor, “Cloud Computing: Revolutionizing Digital Transformation in the Modern Era,” *European Journal of Computer Science and Information Technology*, vol. 13, no. 18, pp. 12–23, 2025, doi:10.37745/ejcsit.2013/vol13n18122.
2. S. Akter, K. Michael, M.R. Uddin, G. McCarthy, and M. Rahman, “Transforming business using digital innovations: The application of AI, blockchain, cloud and data analytics,” *Artificial Intelligence in Operations Management,* vol. 308, pp. 7–39, 2022.
3. S.S. Bhatia, A. Rai, and H. Kaur, “An architectural framework for the implementation of ERP using cloud computing in SMEs: A Literature Survey,” *International Journal of Science and Research,* vol. 6, pp. 11–18, 2017.
4. A. Bentajer, M. Hedabou, K. Abouelmehdi, and S. Elfezazi, “CS-IBE: A Data Confidentiality System in Public Cloud Storage System,” *Journal* of *Procedia Computer. Science*. vol. 141, pp. 559–564, 2018, doi: 10.1016/j.procs.2018.10.126.
5. Y.S. Abdulsalam and M. Hedabou, “Security and Privacy in Cloud Computing: Technical Review,” *MDPI Journal of Future Internet,* vol. 14, no. 11, pp. 1–27, 2022, doi:10.3390/fi14010011.
6. P. Li, J. Huang, T. Li, C.Z. Gao, S.M. Yiu, and K. Chen, “Multi-key privacy-preserving deep learning in cloud computing,” *Future Generation Computer System*, vol. 74, pp. 76–85, 2017, doi:10.1016/j.future.2017.02.006.
7. M.M. Sakr, M.A. Tawfeeq, and A.B. El-Sisi,“Network Intrusion Detection System based PSO-SVM for Cloud Computing,” *International Journal of Computer Network and Information Security*, vol. 3, pp. 22–29, 2019, doi: 10.5815/ijcnis.2019.03.04.
8. A. Patel, M. Taghavi, K. Bakhtiyari, and J. C. JúNior, “ An intrusion detection and prevention system in cloud computing: A Systematic Review,” *Journal of Network and Computer Application*, vol. 36, no. 1, pp. 25–41, 2013, doi: 10.1016/j.jnca.2012.08.007.
9. P. Mishra, E.S. Pilli, V. Varadharadan, and U. Tupakula, “Intrusion Detection Techniques in Cloud Environment: A Survey,” *Journal of Network and Computer Applications*, vol. 77, pp. 18–47, 2017, doi: 10.1016/j.jnca.2016.10.015

10. A. Zarrabi and A. Zarrabi, “Internet Intrusion Detection System Service in a Cloud,” *International Journal of Computer Science Issues*, vol. 9, pp. 308–315, 2012.
11. C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Radaradan, “A Survey of Intrusion Detection Techniques in Cloud,” *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 42–57, 2013, doi: 10.1016/j.jnca.2012.05.003.
12. A. Sari, “A review of anomaly detection systems in cloud networks and survey of cloud security measures in cloud storage applications,” *Journal of Information Security*, vol. 6, no. 2, 142–154, 2015, doi: 10.4236/jis.2015.62015.
13. T. Saranya, S. Sridevi, C. Deisy, T.D. Chung, and M.A. Khan, “Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review,” *Procedia Computer Science*, vol. 171, pp. 1251–1260, 2020, doi: 10.1016/j.procs.2020.04.133.
14. V. Chang, L. Golightly, P. Modesti, Q. A. Xu, L.T. Doan, K. Hall, S. Boddu, and A. Kobusinska. “A Survey on Intrusion Detection Systems for Fog and Cloud Computing,” *Future Internet*, vol. 14, no. 89, pp. 1–27, 2022, doi: 10.3390/fi14030089.
15. S. Mehibs and S. Hashim, “Proposed Network Intrusion Detection System Based on Fuzzy C Mean Algorithm in Cloud Computing Environment,” Journal of Babylon University/Pure and Applied Sciences, vol. 26, pp. 27–35, 2017.
16. A. Handa, A. Sharma, and S.K. Shukla, “Machine learning in cybersecurity: A review,” *WIREs Data Mining Knowledge Discovery*, 2019, doi: 10.1002/widm.1306.
17. H. Gupta and S. Sharma, “Security Challenges in Adopting Internet of Things for Smart Network,” *In Proceedings of the 2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT),* pp. 761–765, 2021.
18. C.E. Cîrnu, C.I. Rotuna, A.V. Vevera, and R. Boncea, “Measures to Mitigate Cybersecurity Risks and Vulnerabilities in Service-Oriented Architecture,” *Stud. Inform. Control*, vol. 27, no. 3, pp. 359–368, 2018, doi: 10.24846/v27i3y201811.
19. J. L. Gutierrez-Garcia, E. Sanchez-DelaCruz, and M. P. Pozos-Parra, “A Review of Intrusion Detection Systems Using Machine Learning: Attacks, Algorithms And Challenges,” *Future of Information and Communication Conference*, pp. 59–78, 2023.
20. N. TN and D. Pramod, “Insider Intrusion Detection Techniques: A State-of-the-Art Review,” *Journal of Computer Information System,* vol. 64, no. 1, 106–123, 2024, doi: 10.1080/08874417.2023.2175337.
21. G. Luo, Z. Chen, and B. O. Mohammed, “A Systematic Literature Review of Intrusion Detection Systems in the Cloud-Based Iot Environments,” *Concurr. Comput. Pract. Exp.,* vol, 34, 2022, doi: 10.1002/cpe.6822.
22. B. Gupta B, D.P. Agrawal, and S. Yamaguchi, “Handbook of Research on Modern Cryptographic Solutions for Computer and Cyber Security,” IGI Global, 2016, doi: 10.4018/978-1-5225-0105-3.
23. E. Vasilomanolakis, S. Karuppayah, M. Muhlhauser, and M. Fischer, “Taxonomy and survey of collaborative intrusion detection.,” *ACM Computing Surveys,* vol. 47, no. 4, 2015, doi: 10.1145/2716260.
24. L. Diana, P. Dini, and D. Paolini, “Overview on Intrusion Detection Systems for Computers Networking Security,” *MDPI Journal of Computers*, vol 14, no 87, pp. 1–44, 2025, doi: 10.3390/computers14030087
25. I.S. Thaseen and C.A. Kumar, “Intrusion Detection Model Using a Fusion Of Chi-Square Feature Selection And Multiclass SVM,” *Journal of King Saud University - Computer and Information Sciences*, vol. 29, no. 4, pp. 462–472, 2017, doi: 10.1016/j.jksuci.2015.12.004
26. P. Ebenezer and A.O. Adewumi, “Efficient Feature Selection Technique for Network Intrusion Detection System Using Discrete Differential Evolution and Decision,” *International Journal of Network Security*, vol. 19, no.5, pp. 660–669, 2017, doi: 10.6633/IJNS.201709.19(5).02.
27. J.R. Beulah and D.S. Punithavathani, “Simple hybrid feature selection for enhancing network intrusion detection with NSL-KDD Dataset,” *International Journal of Applied Engineering Research*, vol. 10, pp. 40498–40505, 2015.
28. S. Sharma and S.Z Hussain, “Weighted Coefficient Firefly Optimization Algorithm and Support Vector Machine For Trust Model And Link Reliability,” International Journal of Computer Networks & Communications, vol. 14, no. 5, pp. 117–132, 2022, doi : 10.5121/ijcnc.2022.14508.
29. S. Bazi, R. Benzid, Y. Bazi, and M.M Al Rahhal, “A Fast Firefly Algorithm for Function Optimization: Application to the Control of BLDC Motor,” *Sensor*, vol. 21, pp. 1–23, doi: 0.3390/s21165267.
30. M.K. Ariyaratne, T.G. Fernando, and S. Weerakoon, “Solving Systems of Nonlinear Equations Using a Modified Firefly Algorithm (modfa),” *Swam and Evolutionary Computation*, vol. 48, pp. 72–92, doi: 10.1016/j.swevo.2019.03.010.