Journal of Information Systems Engineering and Management
2025, 10(18s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

IDS Framework based on ML for Cloud Computing

Prof. Pinal J. Patel[1], Prof. Kapilkumar C Dave[2], Ms. Manisha M. Chaudhari[3], Prof. Hardik Mahendrabhai Patel[4], Prof. Samina Mansuri[5], Prof. Hemangini Shukla[6]

[1]Computer Engineering Department, Government Engineering College, Gandhinagar, Gujarat, India
[2]Intrumentation and Control Engineering Department, Government Engineering College, Gandhinagar, Gujarat, India
[3]Computer Engineering Department, Government Polytechnic Gandhinagar, Gujarat, India
[4]Computer Engineering Department, Vidush Somany Institute of Technology & Research, Kadi, Gujarat, India
[5] Computer Engineering Department, Shankersinh Vaghela Bapu Institute of Technology, Gandhinagar, Gujarat, India
[6]Mathematics - General Department, Government Engineering College, Gandhinagar, Gujarat, India

ARTICLE INFO

Received: 16 Dec 2024

Revised: 25 Jan 2025

Accepted: 10 Feb 2025

ABSTRACT

Introduction: The last few decades have seen a marked increase in individuals' dependence on the internet. Users require considerable internet data and a wide variety of services. Cloud computing offers on-demand services under a "pay-as-you-use" framework. Given its open and distributed nature, security is a paramount concern. An intrusion detection system (IDS) serves the purpose of overseeing activities and identifying any unauthorized access or attacks on the computing system. Machine learning (ML) techniques are effective in identifying both known and unknown threats. In this study, we have suggested an IDS framework that addresses the problem of a single point of failure by utilizing the collaboration between the user and the cloud service provider to operate the IDS. It makes use of clustering followed by classification approaches. Clustering helps in minimizing the data size, time to respond and shorten the training period for classification. We conducted an evaluation of the proposed IDS framework's suitability using KDD cup 1999 dataset on the cloud platform. In the first experiment, clusters were labelled using k nearest neighbour method on cloud VMs users, and all VMs's clusters are merged label wise for classification. In the second experiment, all clusters belonging to the same cloud VM user are aggregated before classifying them. The results show that the low frequent attacks are more accurately detected in second experiment than the first one. The first experiment excels in detecting Probe and Denial of service attacks. However, both experiment tend to have a high detection rate for normal data, exceeding 97%.

Objectives: The objective of this paper is to design and implement IDS framework using machine leanring techniques for cloud platform.

Methods: To evaluate the performance and functionality of the proposed IDS, we have carried out two independent experiments on a cloud platform using the KDD CUP 1999 intrusion dataset. This dataset includes 19.69% normal data, 2% of various attack types such as R2L, U2R, and PROBE and 79.23% DOS attacks. In these experiments, we have utilized 98,804 instances for testing and 395,216 instances for training the models. Labeled clusters of each cloud user – VMs are combined in experment 1. All clusters of a cloud users are combined in experiment 2.

Results: The first scenario performs better at detecting DoS and probe attacks, while the second scenario is more effective at identifying low-frequency attacks. In both scenarios, over 97% of normal data is detected. Additionally, the proposed solution improves the detection rate of DoS and regular attacks by 0.1%.

Conclusions The model is implemented in two stages. In the first stage, a clustering technique is used to reduce reaction time and data volume. This stage helps distinguish between different types of attacks and identify multiclass attacks. The supervised learning approach benefits from reduced training time by utilizing clusters generated from the dataset. Cloud providers and consumers participating in the proposed IDS reduce the risk of a single point of failure. In the event of a virtual machine failure, the IDS remote controller can still detect the intrusion. The applicability of the proposed IDS framework has been verified using the KDD CUP 1999 dataset.
Keywords: Cloud Computing, Clustering, Classification

INTRODUCTION

The World Wide Web became publicly available in 1993 after the development of a browser by Marc Andreesen [1]. The number of individuals connected to the Internet boosted from 16 million in 1995 to 2,937 million by 2014 [2]. The significant growth in the number of internet users has made all businesses available online. As the array of services offered via the Internet expands rapidly, the frequency of attacks on these services also increases. The most of the computer systems possess security vulnerabilities that are both expensive for manufacturers to address and technically complex to resolve. Intrusion can be defined as a compromise of security services like availability, confidentiality, integrity etc. It leads to the significant financial losses and damage the reputation of the organisation that are providing the online services. Therefore, IDS is used to check the activities of the computer and/or network for finding malicious behaviour and generate an alert to the administrator.

Intrusion Detection Systems (IDS) can be classified into two distinct categories based on their detection principles: anomaly detection and misuse detection. The primary goal of misuse detection is to identify known attacks, while anomaly detection aims to detect unknown threats. Misuse detection methods typically utilize pattern matching techniques, which may contain states [3], signatures [4], rules [5], protocols [6] and system calls [7]. In contrast, anomaly detection models monitor the standard behavior of a system to identify any anomalies, triggering alerts when the current behaviour swifts from expected patterns [8] [9]. Moreover, IDS can be classified into interval-based and real-time systems according to their detection period [12]. Interval-based IDS processes and stores data in batches, issuing alerts as each batch is processed, while real-time IDS operates by receiving information instantaneously [10]. Comparing Real-Time Based IDS to Interval-Based IDS, the former is more adept at providing timely responses. In contrast, when the detection rate of a Real-Time Based IDS is insufficient, it may result in the dropping of packets, thereby elevating the false alarm rate of the system [10]. Creating fully autonomous, distributed, and highly adaptable intrusion detection systems significantly depends on their distributed and self-organizing attributes [11]. In May 2014, the Bank of China and the Bank of East Asia were targeted by a Distributed Denial of Service (DDoS) attack, which hindered legitimate users from utilizing their online services at a speed of 7.39 Gbps. Furthermore, on July 31, 2015, customers of Ulster Bank, Royal Bank of Scotland and NatWest were unable to access online banking services due to DDoS attacks [13].

OBJECTIVES

The main objective of this paper is to design framework for intrusion detection using machine learning techniques for cloud platform and to verify the suitability of the proposed framework by conducting the expertiments for identifying the attacks. The suggested IDS framework allows cloud users to participate in intrusion detection.

METHODS

We conducted two experiments on a private cloud set up using OpenNebula. Three physical nodes with CentOS operating systems have been added to the Opennebula server and VMs have been deployed using server on nodes. Figure 1 depicts the screenshot of the cloud after adding nodes. Figure 2 shows the screenshot of created VMs on Cloud.

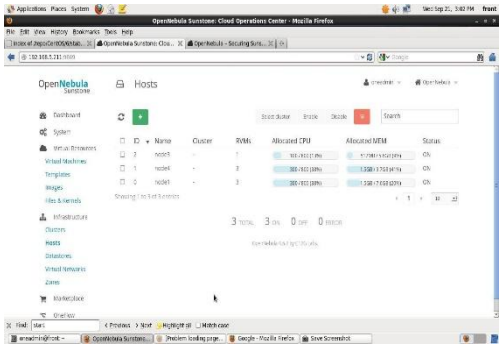

Figure 1: Nodes on Cloud

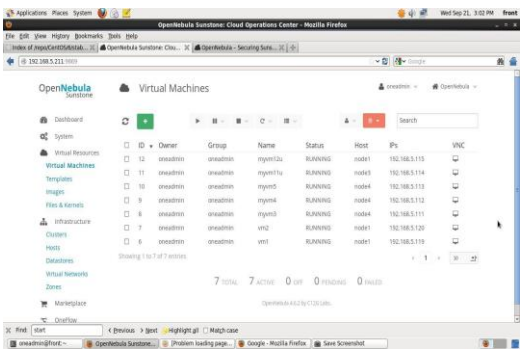

Figure 2: VMs on Cloud

Figure 3 illustrates the proposed system. The subsequent section delineates the procedures of the suggested system.

- The IDS remote controller (IDS- RC) is hosted on cloud providers. When the IDS module service is launched,

all active cloud users - virtual machines (VMs) are prompted to enable IDS, and a 60-second waiting period begins to determine if they reply affirmatively or negatively.

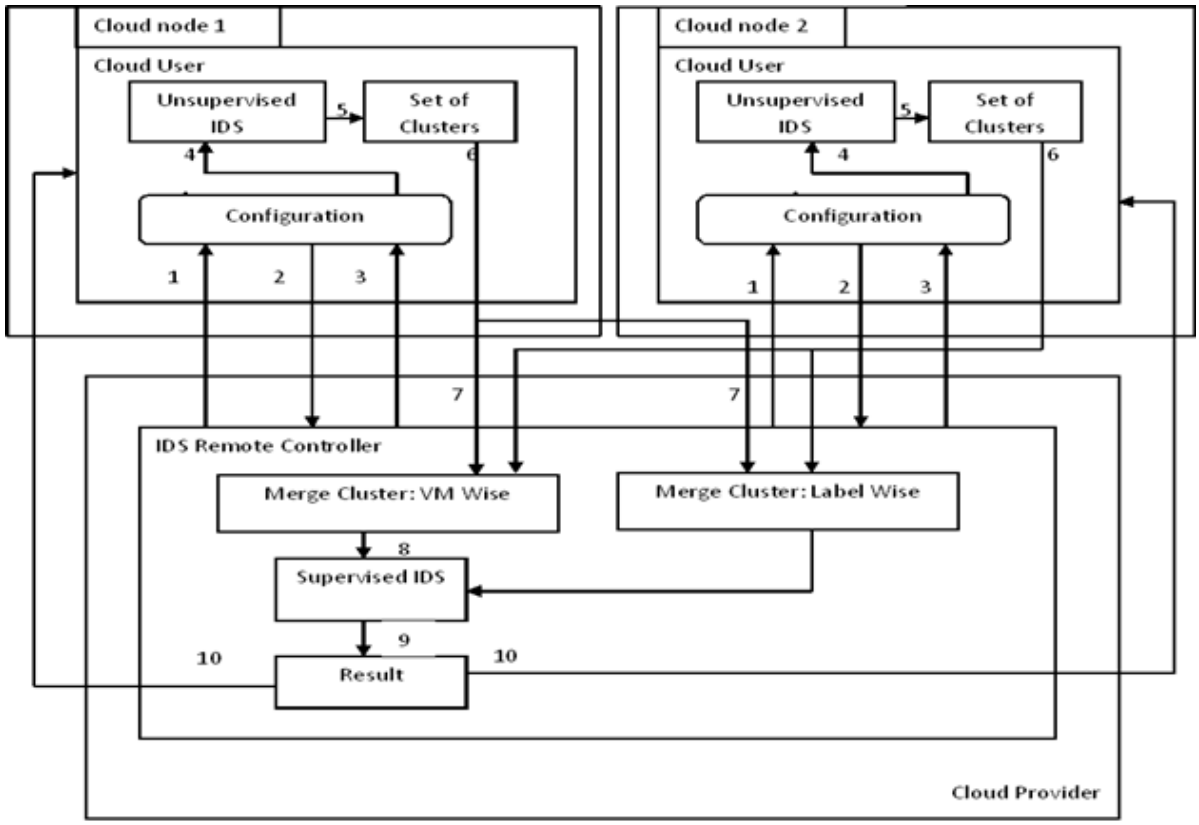

Figure 3: Proposed system

- IDS requires physical resources to function, therefore active VMs receive IDS requests and evaluate their resource capacity. If VMs are ready to run IDS with the enough resources, they will respond positively to the IDS-RC. In contrast, if resources are insufficient, the VMs will deny the participation.
- If active VMs react favorably, the IDS-RC keeps track of their information, starts an IDS instance for each, and waits for cluster data within a predetermined timeout window. The IDS-RC uses a clustering technique to create k clusters in the event that no clusters are received from active VMs.
- k set of clusters are constructed using clustering method on active VMs and send them to the IDS-RC after starting IDS instance on active VMs to anticipate intrusion activities. If clusters are not received within the allotted 60 seconds by IDS-RC, it will decrement the counter of the IDS cloud users - VMs by the quantity of VMs that failed to send clusters.
- The IDS-RC compiles the k clusters it obtains according to the names of the clusters and the virtual machines utilized by cloud users. In particular, it combines clusters from two virtual machines (VM1 and VM2) in one of two ways: (1) by uniting Cluster 0 and Cluster 1 from both VMs, or (2) by pairing Cluster 0 from VM1 with

Cluster 0 from VM2, and Cluster 1 from VM1 with Cluster 1 from VM2. Subsequently, the data is categorized as either normal or indicative of an intrusion with corresponding attack labels through the classification method.

- The IDS-RC issues alerts to all cloud user - virtual machines associated with the IDS module, based on the identified findings.

RESULTS

1. Case 1: Labeled clusters of each cloud user – VMs are combined. The result of the case 1 is shown in Table 1.
2. Case 2: All clusters of a cloud users are combined. Table 2 illustrates the outcome foe case 2.

Figure 4 and figure 5 show the accuracy curve over epochs for case 1 and case 2 respectively.

Table 1 Results of Case 1

Result of Unsupervised – Clustering Method				
True Negative Rate (%)	True Positive Rate (%)	Precision (%)	F1 Score (%)	Accuracy (%)
98	99	93	96	98
Result of supervised – Classification Method				
Types of Attacks	Probe	U2R	R2L	DoS
Detected Accurately in %	92.4	87.5	64.8	97.9
Precision (%)	92.9	88.2	77.7	99.5
F1 Score (%)	87.7	58.3	82.5	98.9
Recall (%)	84.03	44.5	89.02	99.9

Table 2 Results of case 2

Result of Unsupervised – Clustering Method				
True Negative Rate (%)	True Positive Rate(%)	Precision (%)	F1 Score (%)	Accuracy (%)
98.65	97.7	94.69	96.17	98.46
Result of supervised – Classification Method				
Types of Attacks	Probe	U2R	R2L	DoS
Detected Accurately in %	89.4	90.2	87.5	95.9

Precision (%)	87.74	79.6	95.16	97.2
F1 Score (%)	44.4	63.4	60.14	97.9
Recall (%)	30	53	44.4	99.9

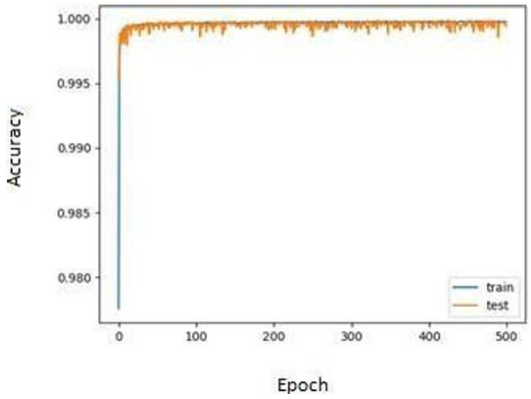

Figure 4: Accuracy Curve for Case 1

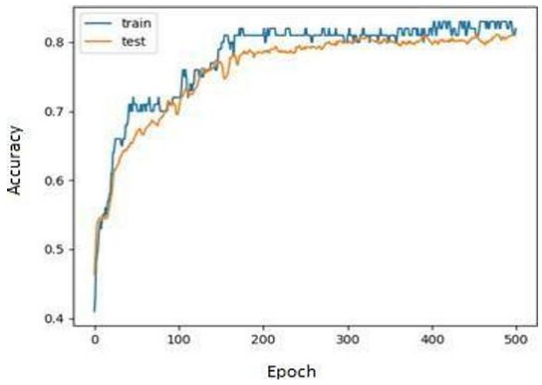

Figure 5: Accuracy Curve for Case 2

DISCUSSION

The model is implemented in two stages. In the first stage, a clustering technique is used to reduce reaction time and data volume. This stage helps distinguish between different types of attacks and identify multiclass attacks. The supervised learning approach benefits from reduced training time by utilizing clusters generated from the dataset. Cloud providers and consumers participating in the proposed IDS reduce the risk of a single point of failure. In the event of a virtual machine failure, the IDS remote controller can still detect the intrusion. The applicability of the proposed IDS framework has been verified using the KDD CUP 1999 dataset. The first scenario performs better at detecting DoS and probe attacks, while the second scenario is more effective at identifying low-frequency attacks. In both scenarios, over 97% of normal data is detected. Additionally, the proposed solution improves the detection rate of DoS and regular attacks by 0.1%.

REFRENCES

[1] Prezi (2016) The Internet. Available https://prezi.com/nt0kdsbj7ijg/the-internet/ [Accessed in December 2016]

[2] Mayo K., Newcomb P (2008) The birth of the world wide web: An oral history of the Internet. http://www.vanityfair.com/news/2008/07/internet200807 [Accessed in December 2011]

[3] K. Ilgun (1993) USTAT: A real-time intrusion detection system for UNIX, Proceedings of IEEE Symposium on Security and Privacy pp 16–28

[4] J. Peng, C. Feng, and J. Rozenblit (2006) A hybrid intrusion detection and visualization system, Proceedings of 13th Annual IEEE International Symposium and Workshop on Engineering of Computer Based Systems pp 505–506

[5] M. Sebring, E. Shellhouse, M. Hanna, and R. Whitehurst (1988) Expert systems in intrusion detection: A case study, Proceedings of 11th National Computer Security Conference, pp 74–81

[6] O. Depren, M. Topallar, E. Anarim, and M.K. Ciliz (2005) An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks, Expert systems with Applications 29 (4) pp 713– 722

[7] Seo, Jeongseok, and Sungdeok Cha (2007) Masquerade detection based on SVM and sequence-baseuser commands profile, Proceedings of 2nd ACM symposium on Information, computer and communications security

[8] Sho Ohtahara,Takayuki Kamiyama (2009) Anomaly-based Intrusion Detection System Sharing Normal Behavior Databases among Different Machines, 9th IEEE International Conference on Computer and Information Technology Xiamen China Volume 1

[9] D. Anderson, T.F. Lunt, H. Javitz, A. Tamaru, and A. Valdes (1995) Detecting unusual program behavior using the statistical component of Next-generation Intrusion Detection Expert System (NIDES) SRI International Computer Science Laboratory

[10] Khan, Nabeel, Younus, Bilal Rauf, and Kabeer Ahmed (2010) Comparative study of intrusion detection system and its Recovery mechanism, The 2nd International Conference on Computer and Automation Engineering (ICCAE) Volume 5

[11] Andrew Hay, Daniel Cid (2008) Chapter Getting Started with OSSEC, book OSSEC Host-Based Intrusion Detection Guide pp 1-27

[12] Pinal J. Patel (2020). Intrusion detection system using machine learning techniques in cloud computing. Gujarat Technological University, Research Repository on Amazon. https://s3-ap-southeast1.amazonaws.com/gtusitecirculars/uploads/Synopsis-Patel%20Pinal 129990907010_4460 69.pdf

[13] Honan B. (2015) DDoS attacks take down RBS, Ulster bank, and NatWest online systems. http://www.csoonline.com/article/2955693/cyber-attacks-espionage/ddosattacks-take-down-rbs-ulster-bank -and-natwest-online-systems.html [Accessed in May 2015]