International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

User Login Behaviour Analysis in HPC clusters using Data Analysis and Probabilistic Technique

Atharv Nagarikar*[1], Abhishek Patel[2], Krishan Gopal Gupta[3], Shashank Sharma[4], Mohammed Afzal[5], Sanjay Wandhekarl[6]

Submitted: 21/08/2023 **Revised**: 08/10/2023 **Accepted**: 21/10/2023

Abstract: The login behaviour of users in HPC clusters will be examined in this research paper with the use of machine learning, data analysis and probabilistic techniques to identify patterns that can be used to identify anomalous login behaviour based on IP and timings. The customized probabilistic model will monitor each user's login behaviour and stop any unauthorized activity in HPC clusters.This paper also discusses the data preprocessing and feature engineering techniques to extract information from sshd logs. Despite several attempts to utilize machine learning models to recognize and save user patterns from sshd logs, the majority of the models proved unsuccessful. Since the sshd logs do not exhibit any distinct login behaviour or pattern of users, machine learning models trained on specific behaviour or patterns from datasets struggle to detect even slight changes in user login behaviour, resulting in a high rate of failure for such models. One major factor contributing to the failure of machine learning models is the highly imbalanced dataset consisting solely of true values, which makes it challenging for the models to identify outliers or patterns. Additionally, since the users do not follow any specific pattern, it may be necessary to establish a single pattern and categorize the login behaviour of all users accordingly to achieve successful results. To overcome the challenges of identifying user login patterns and detecting anomalies, an automated approach is necessary. This approach should involve the development of a system that can detect and store each user's login patterns as probability metrics along with IP addresses. These metrics can then be utilized to predict any anomalous activity based on the identified patterns

Keywords: *Data Analysis, Feature Engineering, HPC clusters, HPC security, Machine Learning, Probabilistic Modeling*

1. Introduction

High-Performance Computing(HPC)[1] involves leveraging supercomputers[2] or computer clusters to handle complex computational problems that require significant processing power and memory. HPC systems are typically large-scale computing resources that involve complex architectures and networks. They are used for a variety of applications, including scientific research, engineering, and financial modeling. These systems possess the capability to process vast amounts of data and perform millions of calculations per second, making them an ideal solution for compute-intensive tasks such as scientific simulations and data analysis.

HPC has revolutionized the way scientists and researchers conduct simulations, modeling, and analysis, providing them with the ability to work at a scale and speed that was previously unattainable. However, as the usage of HPC systems has increased, concerns surrounding their security have also emerged. HPC security[3] pertains to the measures taken to safeguard the systems, networks, and data used in these environments against cyber threats., The very complexity of these systems makes them vulnerable to cyber-attacks. HPC systems are often targeted by system attackers and other malicious actors seeking to steal data disrupt operations, or gain unauthorized access.

To ensure the confidentiality, integrity, and availability of data and systems, HPC security employs various technologies, processes, and policies. Some significant areas of concentration for HPC security include Access Controls, network security, Physical Security, Data Security and Incident Response. The primary emphasis of the paper is on controlling access. Access controls, which restrict access to HPC systems and data based on the principle of least privilege. This implies that users are only provided access to the resources necessary to accomplish their work, and access controls can involve user authentication, authorization, and audit logging.

The login nodes in HPC clusters generate Secure Shell Daemon, or sshd logs. It is a process that silently monitors all user login and authentication attempts in HPC clusters. The systems' unauthorized login attempts are also recorded in these logs. The purpose of this paper is to employ Machine Learning, data analysis[5] and probabilistic techniques to examine the sshd logs and identify patterns in user login behaviour, with the goal of mitigating unauthorized activity.

Data analysis and machine learning are two interconnected

[1]*Senior Project Engineer, HPC-Technologies, C-DAC, Pune*
[2]*Project Engineer, HPC-Technologies, C-DAC, Pune*
[3]*Joint DirectorC-Technologies, C-DAC, Pune*
[4] *Prinicpal Technical Officer, HPC-Technologies, C-DAC, Pune*
[5]*Module Leader, HPC-Technologies, C-DAC, Pune*
[6]*Senior Director HOD, HPC-Technologies, C-DAC, Pune*

fields that have revolutionized the way we make sense of data. Data analysis is the process of inspecting, cleaning, transforming, and modeling data to extract useful insights and information. Machine learning, on the other hand, is a subset of artificial intelligence that involves the development of algorithms that enable machines to learn from data without being explicitly programmed. By leveraging data analysis techniques, machine learning

algorithms can learn from large datasets that have automated decision-making processes, improved accuracy, and increased efficiency.

This paper covers the methods used for converting sshd logs into tabular or CSV format through data preprocessing techniques[6]. Several machine learning models were tested in an experiment to predict the desired outcomes. In addition, the paper describes a probabilistic method for identifying user login behaviour in the cluster.

This research paper presents a comprehensive study of related work in the field of HPC Security. The approach proposed in this paper is discussed, and the various stages of the research are detailed. Firstly, the process of collecting the dataset is described, along with information about its composition. Next, the preprocessing steps taken to clean and transform the data are presented. The use of the HPC cluster for performing the experiment is also explained. Additionally, a custom probabilistic model is introduced, and its working is explained. The execution flow of the experiment is discussed, and details of the evaluation metrics used to assess the results are provided. Finally, the findings and inferences drawn from the experiment are presented, along with suggestions for future work in this area.

2. Related work

In "Security of HPC Systems: From a Log-analyzing,"[7] the authors discuss security threats in HPC systems, such as insider and DDoS attacks, and the importance of log analysis in detecting them. Real-world examples are provided, and the paper suggests the need for further research to enhance HPC system security. "User behaviour control method for HPC system"[8] proposes a method for controlling user behaviour in HPC systems by analyzing user behaviour data and using machine learning to detect anomalous behaviour patterns. This approach can improve security and performance in HPC systems, but more research is needed in this area. "Enhancing HPC security with a user-based firewall"[9] suggests using a user-based firewall for better security in high-performance computing by restricting network traffic based on individual user needs and permissions. The paper explains the benefits of this approach and provides a proof-of-concept implementation. "Autonomous Security Mechanisms for High-Performance Computing Systems" [10]analyzes various autonomous security mechanisms for HPC systems and evaluates their effectiveness. The paper suggests future research in this field to address the challenges of securing HPC systems. "Security in high-performancecomputing environments"[11] examines security concerns in HPC and proposes ways to reduce risks while maintaining high performance. Topics such as network security, user authentication, and data protection are discussed, and the paper explores the challenges and trade-offs in implementing security measures in HPC systems.

3. Proposed Approach

3.1. HPC cluster

A High-Performance Computing (HPC) cluster is a type of computing cluster that consists of multiple interconnected computers or servers that work together to perform high-performance computing tasks. HPC clusters are used to solve complex computational problems that require a significant amount of computing power, such as scientific simulations, weather forecasting, molecular modeling, and financial analysis.

HPC clusters typically consist of a large number of interconnected nodes or computing servers, each of which is equipped with a high-speed network interface, high-speed processors, and large amounts of memory. These nodes are typically housed in a single location, such as a data center, and are connected together through a high-speed network.

The high-speed network is a critical component of an HPC cluster, as it enables the nodes to communicate and share data quickly and efficiently. In addition to the network, an HPC cluster also requires specialized software and management tools to effectively distribute computing tasks across the nodes and manage the overall system.

HPC clusters can be designed and configured to meet the specific needs of different applications and workloads. For example, some clusters may be optimized for scientific computing tasks, while others may be designed for data analytics or machine learning workloads.

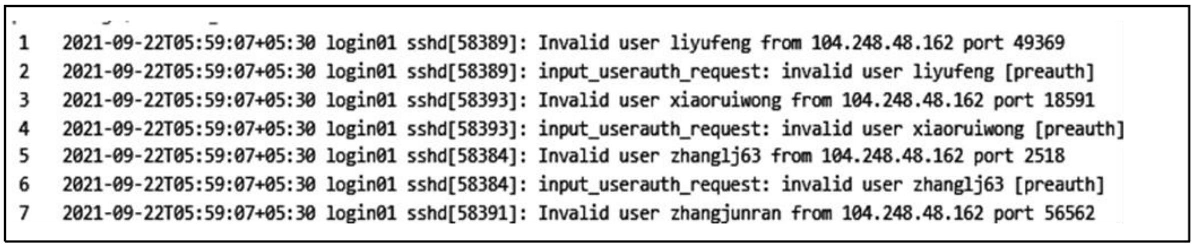
3.2. Dataset

SSH (Secure Shell) is a network protocol that allows secure remote access to a computer system. SSHD is the SSH server daemon that runs on the server side to provide the SSH service.

In HPC (High-Performance Computing) clusters, SSHD logs can be an important source of information for system administrators to monitor and troubleshoot system activity. SSHD logs record all SSH connections to the HPC cluster, including the user who made the connection, the source IP address, the time and date of the connection, and the outcome of the connection attempt (success or failure).

By analyzing SSHD logs, system administrators can identify potential security threats, such as failed login attempts or unauthorized access attempts, and take appropriate measures to protect the system. They can also track user activity and resource usage to optimize system performance and ensure fair allocation of resources.

Overall, SSHD logs provide valuable insight into system activity and can help HPC cluster administrators maintain a secure and efficient computing environment Figure 1.



```
1   2021-09-22T05:59:07+05:30 login01 sshd[58389]: Invalid user liyufeng from 104.248.48.162 port 49369
2   2021-09-22T05:59:07+05:30 login01 sshd[58389]: input_userauth_request: invalid user liyufeng [preauth]
3   2021-09-22T05:59:07+05:30 login01 sshd[58393]: Invalid user xiaoruiwong from 104.248.48.162 port 18591
4   2021-09-22T05:59:07+05:30 login01 sshd[58393]: input_userauth_request: invalid user xiaoruiwong [preauth]
5   2021-09-22T05:59:07+05:30 login01 sshd[58384]: Invalid user zhanglj63 from 104.248.48.162 port 2518
6   2021-09-22T05:59:07+05:30 login01 sshd[58384]: input_userauth_request: invalid user zhanglj63 [preauth]
7   2021-09-22T05:59:07+05:30 login01 sshd[58391]: Invalid user zhangjunran from 104.248.48.162 port 56562
```

Figure 1:- Example of sshd logs.

3.3. SSHD Log Collection

HPC clusters are collections of multiple login and compute. So a user who wants to login, can land at any login node. In order to generate data for preprocessing it is required to collect all the data in one file. This process should be automated in such a way that every minute generated logs get accumulated in file Figure2.

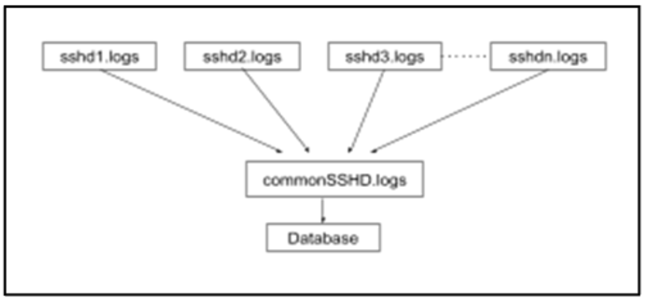


Fig 2:- Automated process for accumulating sshd logs.

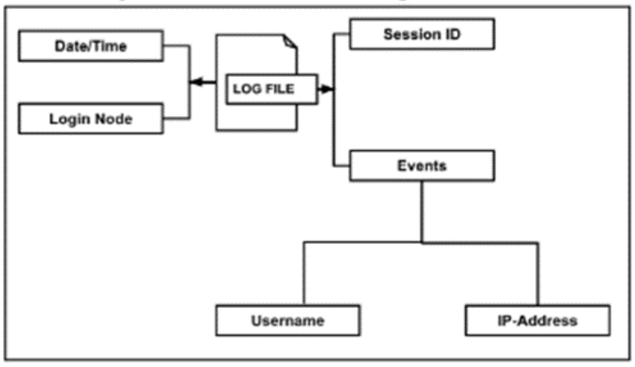


Figure 3:- Parameters in sshd logs

3.4. Preprocessing and Analysis of SSHD Log collection

In the log file, there are four features: date and time, login node, SSHD session ID, and events [Figure 3]. These features capture what event is done by the user at a particular time. In these events, In these events,the user information, such as username and IP address, is present. As this data is in text format, we applied regular expressions to filter out this data from the log file. From each line of log data, we have filtered date and time, login node, and session ID. From events, we have filtered username and IP address. We then converted all of these features into a Pandas DataFrame. Each parameter is explained in the following points.

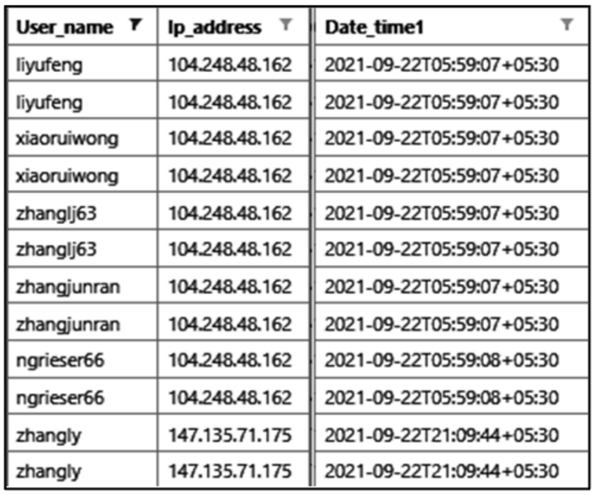
1) 2019-10-19T17:07:58-04:00- Date and time of occurring events in YYYY-MM-DD T HH:MM:SS format.

2) login01 - login node detail

3) sshd[85037]- session ID of a user

4) input_userauth_request: invalid user axxx [preauth]- These are events of the user, It includes some important information of user login details

In the preprocessing phase, the sshd logs were parsed to extract relevant information such as the username, IP address, and date-time [Figure 4].



User_name	Ip_address	Date_time1
liyufeng	104.248.48.162	2021-09-22T05:59:07+05:30
liyufeng	104.248.48.162	2021-09-22T05:59:07+05:30
xiaoruiwong	104.248.48.162	2021-09-22T05:59:07+05:30
xiaoruiwong	104.248.48.162	2021-09-22T05:59:07+05:30
zhanglj63	104.248.48.162	2021-09-22T05:59:07+05:30
zhanglj63	104.248.48.162	2021-09-22T05:59:07+05:30
zhangjunran	104.248.48.162	2021-09-22T05:59:07+05:30
zhangjunran	104.248.48.162	2021-09-22T05:59:07+05:30
ngrieser66	104.248.48.162	2021-09-22T05:59:08+05:30
ngrieser66	104.248.48.162	2021-09-22T05:59:08+05:30
zhangly	147.135.71.175	2021-09-22T21:09:44+05:30
zhangly	147.135.71.175	2021-09-22T21:09:44+05:30

Fig 4:- Extracted Data

3.5. HPC cluster for Experiment

The Param Sanagank HPC cluster systems developed by CDAC, were used to conduct the execution experiment. The systems utilized in this project are equipped with Intel Xeon Platinum 8268 processors and NVIDIA Tesla V100 GPUs boasting a combined peak performance of 1.6 PFLOPS. The cluster consists of a total of 332 nodes, with eight of these nodes designated as login nodes[12].

3.6. Custom Probabilistic Model

The study involved experimenting with several Machine Learning models to identify and store user patterns from the sshd logs, but most of them failed to produce desirable results. Machine Learning models are typically trained on specific patterns or behaviours from datasets. However, the sshd logs lack any particular login behaviour or user pattern, causing most of the Machine Learning models to fail in detecting even slight changes in user login behaviour [Figure 5]. Furthermore, the dataset only contains true values and is highly imbalanced[13], making it more challenging for typical Machine Learning models to identify outliers or patterns. Given the absence of any consistent user behaviour, it becomes necessary to establish a single pattern and categorize the login behaviour of all users according to that pattern. Probabilistic modeling offers a solution to address the limitation of deterministic modeling. While deterministic models assume that all variables and inputs are known and certain, probabilistic models account for uncertainty and variability in the data. This enables probabilistic models to provide more realistic and accurate predictions and outcomes, making them particularly useful when dealing with complex systems or situations where multiple variables are involved[14]. By using probabilistic modeling, it becomes possible to project multiple possible outcomes, taking into account the variability and uncertainty of the data. This approach allows for the identification of potential risks and the exploration of alternative scenarios, providing a more comprehensive and realistic understanding of the system being modeled. To effectively store user login patterns, it may be beneficial to categorize users based on their typical login behaviour or the time period during which they usually log in. This can help to establish a baseline for each user's login pattern and allow for the calculation of login probability during each time period. By storing this information in the form of metrics, it becomes possible to analyze and track user behaviour over time, identify any deviations from their typical patterns, and take appropriate action as needed.

Machine Learning Model	**Evaluation Metric (accuracy)**
K-means Cluster	0.10
SVM(kernel rbf)	0.88
One Class SVM	0.431
XGBoost	0.60

Fig 5:- Evaluation Metric results of Machine Learning models

3.6.1. Training Custom probabilistic Model

The custom model accepts the processed .csv file, the processed .csv file consists of features named ‘Date_time’,’user_name’,’Ip_address’,‘login_node’ number. The process of the custom model involves handling data on a per-user basis. The model first the sort the data according to the date and analyze the login behaviour of user as per time and categorized the behaviour into six parts i.e. mid morning (6am to 8am), morning (8am to 12pm), afternoon (12 pm to 4pm), evening(4pm to 8 pm), mid night(8pm to 12 am) and night (12 am pm to 6am) based on date_time feature. Then the model calculates the probability for each category.

Let's represent the processed .csv file as a matrix with dimensions n x 4, where n is the number of rows in the file. Each row represents a user's login information and has the following columns: Date_time, user_name, Ip_address, login_node.

Let D_i represent the date and time of login for the i-th row in the matrix. We can use the following formula to categorize the login behaviour of a user:

B(D_i) = mid morning, if 6am ≤ D_i < 8am B(D_i) = morning, if 8am ≤ D_i < 12pm B(D_i) = afternoon, if 12pm ≤ D_i < 4pm B(D_i) = evening, if 4pm ≤ D_i < 8pm B(D_i) = mid night, if 8pm ≤ D_i < 12am B(D_i) = night, if 12am ≤ D_i < 6am

where B(D_i) represents the login behaviour of the user

Now, let's define P(b) as the probability of a user logging in during a specific time category b, where b can take the values of mid morning, morning, afternoon, evening, mid night, or night. The probability can be calculated as:

P(b) = (number of logins in time category b) / (total number of logins for the user)

The probability of these login time categories along with IP addresses were stored in the form of a .csv file as user metric [Figure 6].

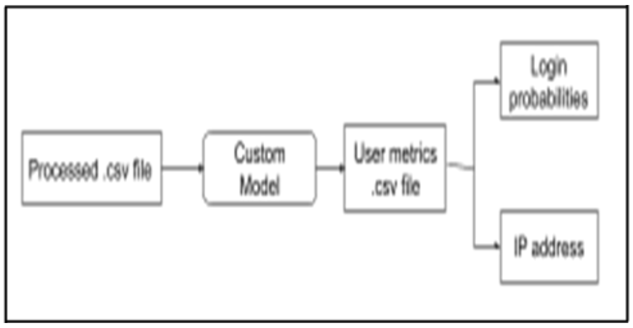
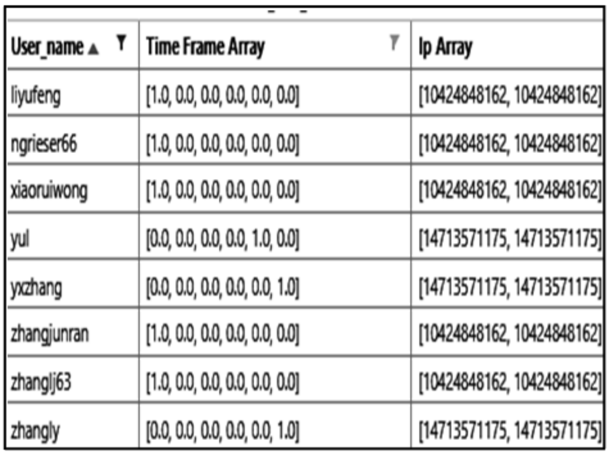


Fig 6:- Execution of Custom model



User_name	Time Frame Array	Ip Array
liyufeng	[1.0, 0.0, 0.0, 0.0, 0.0, 0.0]	[10424848162, 10424848162]
ngrieser66	[1.0, 0.0, 0.0, 0.0, 0.0, 0.0]	[10424848162, 10424848162]
xiaoruiwong	[1.0, 0.0, 0.0, 0.0, 0.0, 0.0]	[10424848162, 10424848162]
yul	[0.0, 0.0, 0.0, 0.0, 1.0, 0.0]	[14713571175, 14713571175]
yxzhang	[0.0, 0.0, 0.0, 0.0, 0.0, 1.0]	[14713571175, 14713571175]
zhangjunran	[1.0, 0.0, 0.0, 0.0, 0.0, 0.0]	[10424848162, 10424848162]
zhanglj63	[1.0, 0.0, 0.0, 0.0, 0.0, 0.0]	[10424848162, 10424848162]
zhangly	[0.0, 0.0, 0.0, 0.0, 0.0, 1.0]	[14713571175, 14713571175]

In Figure 7, represents the user metric generated by the model.

This metric was specifically designed to capture and quantify user behaviour or engagement within the context of the system along with IP address.

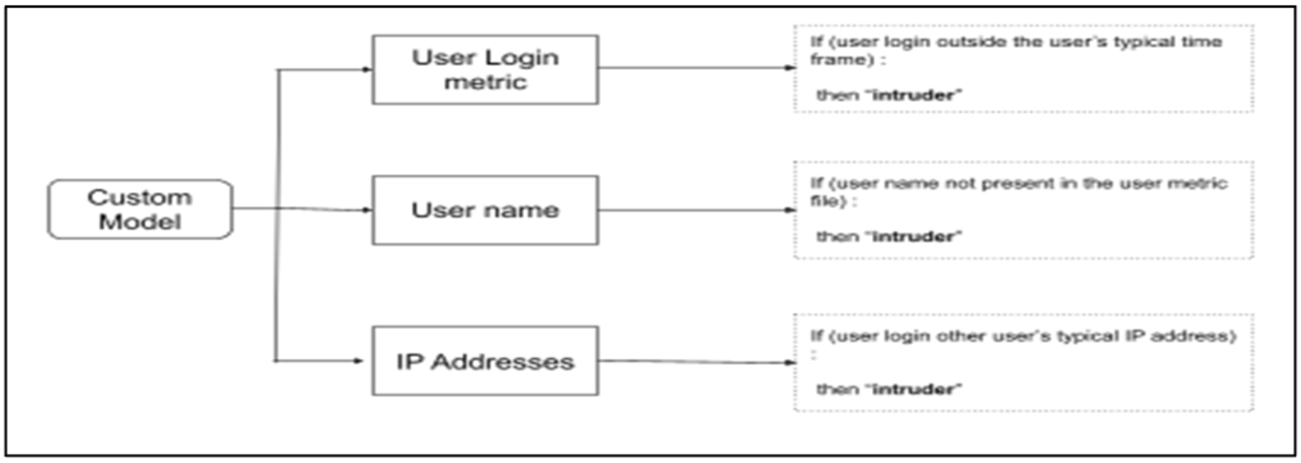

Custom Model
User Login metric
User name
IP Addresses
If (user login outside the user's typical time frame) : then "intruder"
If (user name not present in the user metric file) : then "intruder"
If (user login other user's typical IP address) : then "intruder"

Fig 7:- User Metric

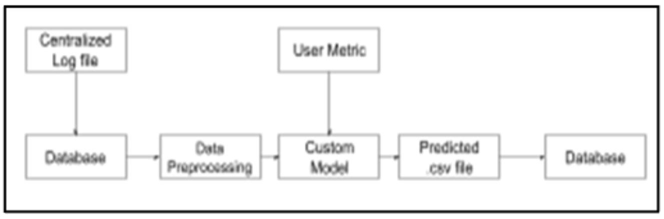


Figure 8:- parameters are evaluated by the custom model to provide a judgment

3.6.2. Model Prediction

The model predicts the output based on three parameters i.e. user name, user login attempts and IP addresses used by user.The output generated by the model is designed to detect any attempts at logging in outside of the user's typical time frame. In such cases, the model will quickly identify the change in login patterns as the elevant login category will update from a value of zero to a probability value. Since the model has already established a metric for the IP address feature, it will determine if the user attempts to log in using their regular IP addresses. If the user logs in using a different IP address, the model will classify the user as an intruder [Figure 8].

The model generates an output value denoted as P, which serves as an indicator of the likelihood of detecting an attempt at logging in outside of a user's typical login behaviour. This prediction is based on three key parameters:

1. User Name (U): The presence or absence of the user's name in the generated user metric.
2. IP Address (IP): Whether the user's IP address is included in the generated user metric.
3. Login Activity (A): Whether the user's login attempts occur within the usual time frame established by their historical login patterns.

The model classifies users into two categories:

P = 1 (Intruder): This classification is made when the user's name (U) is not found in the generated user metric, the user's IP address (IP) is not present in the generated user metric, and the user's login activity (A) deviates from their typical login time frame.

P = 0 (Not an Intruder): This classification is applied when the user's name (U) is included in the generated user metric, the user's IP address (IP) is found in the generated user metric, and the user's login activity (A) occurs within their usual login time frame.

In summary, the model provides a probability value P to assess the likelihood of intrusion attempts based on user-specific metrics and login patterns. The model's prediction flags whether each of the three parameters (U, IP, A) contributes to the determination of whether a user is classified as an intruder or not.

[Figure 12].

3.7. Execution Flow

Following Steps were involved in the execution [Figure 9]:-

Figure 9:- Execution Flow

1. The sshd logs from all the login nodes are consolidated and stored in the centralized file
2. The database logs undergo processing and are transformed into a tabular format or .csv file
3. The preprocessed data, in addition to user metrics, serves as the input for the custom model.
4. The model generates predictions based on user metrics and converts the output into a tabular format for each

user in the input data.

5. The model also provides information based on which it predicts the result.
6. The resulting output is stored in the database to be utilized for future use cases and analysis.

3.8. Evaluation Metric

The effectiveness of a model at performing a specific task is measured using the evaluation metrics. These metrics are employed to gauge the model's effectiveness and pinpoint areas in need of development. Because the paper addresses imbalanced data, F1-score is a suitable evaluation metric[15][16]. It is a weighted average of precision and recall, where precision is the ratio of true positives to the total number of predicted positives, and recall is the ratio of true positives to the total number of actual positives.

The formula for F1 score is:

F1 score = 2 * ((precision * recall) / (precision + recall))

Precision: measures the proportion of true positives (correctly identified instances) among the total number of instances identified as positive.

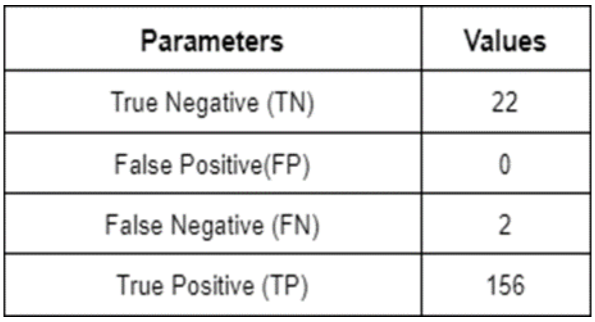
Precision = (True Positives) / (True Positives + False Positives)

Recall measures proportion of actual positives was identified correctly

Precision = (True Positives) / (True Positives + False Negatives)

To obtain the F1-score, a custom test dataset was created consisting of a total of 180 lines. Among these lines, 158 are user data that do not follow their typical pattern, while the remaining 22 lines are user data that adhere to their login pattern. During the inference stage, the model predicted 156 out of 158 lines to be intruders (True Positive) while wrongly classifying 2 actual intruders as non-intruders (False Negative). Additionally, the model correctly identified all 22 lines of users following their usual pattern as non-intruders (True Negatives) [Figure 9 and Figure 10].

The parameters for confusion matrix are as follows:-



Parameters	Values
True Negative (TN)	22
False Positive(FP)	0
False Negative (FN)	2
True Positive (TP)	156

Fig 9:- Parameters for calculating F1-score

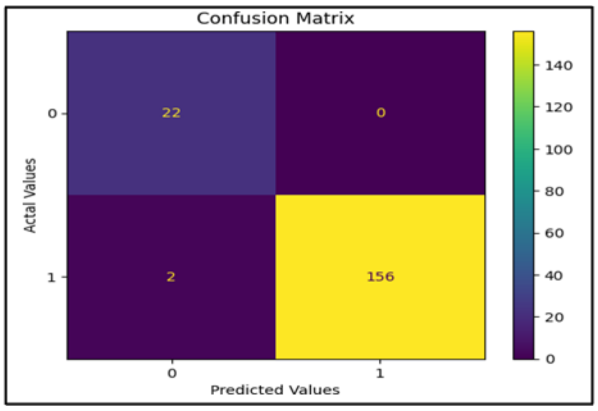
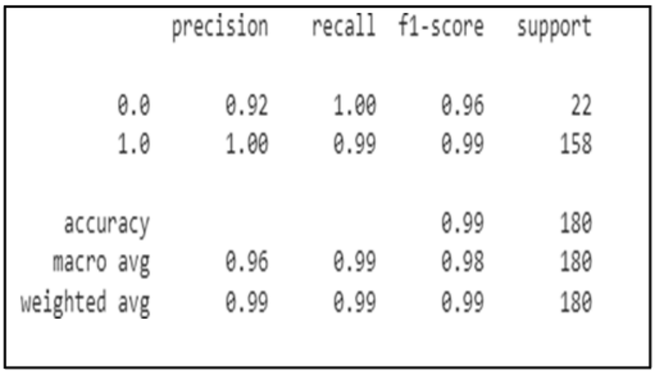


Fig 10:- Confusion Matrix

From the confusion matrix, F1- score, precision and recall is calculated and show below using classification report :-



```
              precision    recall  f1-score   support

         0.0       0.92      1.00      0.96        22
         1.0       1.00      0.99      0.99       158

    accuracy                           0.99       180
   macro avg       0.96      0.99      0.98       180
weighted avg       0.99      0.99      0.99       180
```

Fig 11:- Classification Report

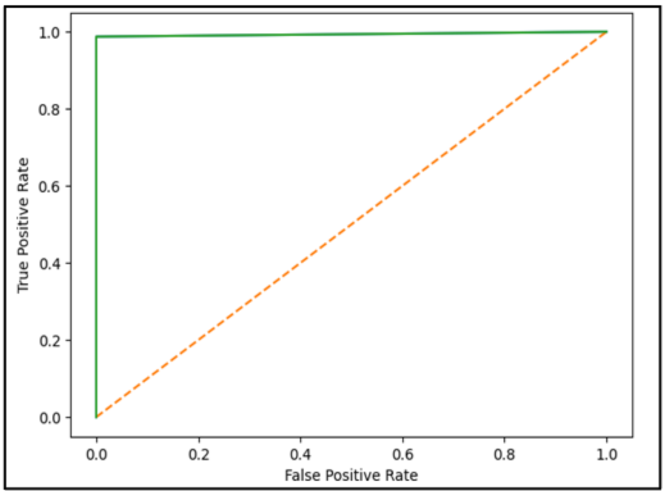


Fig 12 :- AUC-ROC Curve

4. Result and Inference

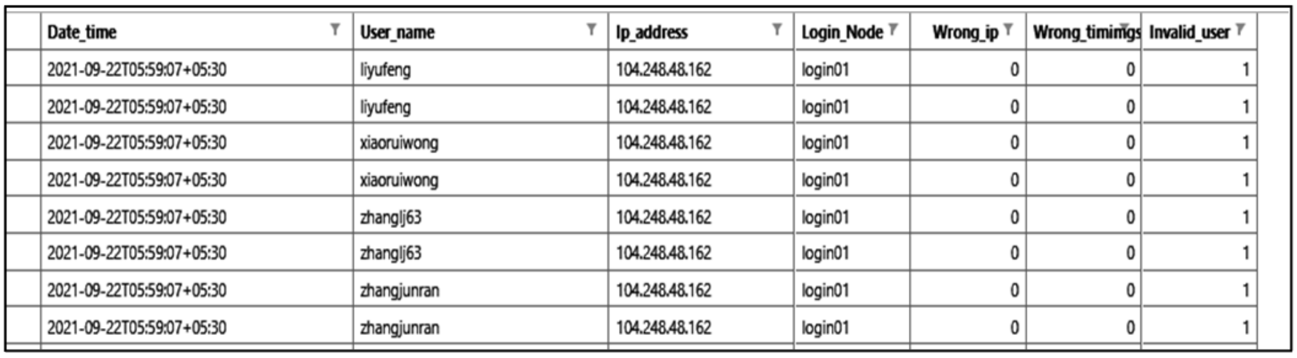
The cluster was effectively monitored by the probabilistic model, which successfully detected intruding activity . The model's True Positive Rate (TPR or Recall) was highly accurate, as evidenced by the confusion matrix. The model was able to correctly identify the majority of the intruding activity on the cluster. Additionally, the AUC-ROC curve [Figure 12] indicated that the model was capable of accurately distinguishing between intruder and non-intruder users in the dataset. The model's performance on the test data was impressive, achieving an f1-score of 0.99 [Figure 11], which indicates that it was highly effective in predicting whether a user was behaving anomalously or not.

These results suggest that our model is reliable for detecting anomalous users with great precision, making it useful for detecting security breaches or other issues in various industries. However, further testing and refinement may be necessary to optimize its performance for specific use cases. Figure 13 provides a snapshot of the test data containing information about intruders, including their usernames, IP addresses, login nodes, and the time of their activity [Figure 13].

5. Future Work

The security of HPC systems is a critical concern, particularly in today's era of increased cyber threats. As the use of machine learning techniques becomes more prevalent, HPC systems will face more complex security challenges. Defending against these attacks requires the development of advanced methodologies that can effectively analyze large volumes of log files and identify potential security threats in real-time. Although several defense strategies have been proposed, most of them only focus on a small portion of the log files, leaving room for improvement. By developing a comprehensive approach that can analyze all available logs and incorporate advanced machine learning techniques, researchers can explore new frontiers in HPC system security and significantly enhance their ability to detect and mitigate potential security threats.

Deep learning methods are widely used in log file analysis to classify different types of users by their application behaviours. It is important to understand that different platforms will have different user behaviours and have different data types, thus leading to different feature extraction . The effective way to use it is to use deep learning techniques.



Date_time	User_name	Ip_address	Login_Node	Wrong_ip	Wrong_timings	Invalid_user
2021-09-22T05:59:07+05:30	liyufeng	104.248.48.162	login01	0	0	1
2021-09-22T05:59:07+05:30	liyufeng	104.248.48.162	login01	0	0	1
2021-09-22T05:59:07+05:30	xiaoruiwong	104.248.48.162	login01	0	0	1
2021-09-22T05:59:07+05:30	xiaoruiwong	104.248.48.162	login01	0	0	1
2021-09-22T05:59:07+05:30	zhanglj63	104.248.48.162	login01	0	0	1
2021-09-22T05:59:07+05:30	zhanglj63	104.248.48.162	login01	0	0	1
2021-09-22T05:59:07+05:30	zhangjunran	104.248.48.162	login01	0	0	1
2021-09-22T05:59:07+05:30	zhangjunran	104.248.48.162	login01	0	0	1

Fig 13: Inference output

As technology continues to evolve, we have witnessed the emergence of various distributed computing models, including cloud computing, cluster computing, and grid computing. These systems are designed to provide efficient and scalable computing solutions to a wide range of applications. However, like HPC systems, they also face similar problems, including security vulnerabilities. Nonetheless, the similarities in their architectures and utilities make it possible to transfer security solutions across these systems.As such, there is an opportunity to leverage existing security solutions in other distributed systems to improve the security posture of HPC systems. This presents an exciting future research topic that could lead to innovative security solutions for HPC systems.

As HPC systems continue to grow in scale and complexity, the security of these systems has become a major concern for both academia and industry. Traditional security measures are often insufficient to protect against sophisticated attacks that are specifically targeted towards HPC systems. However, with the advancements in machine learning, there is an opportunity to enhance the security of HPC systems through the application of machine learning-based security inspection, evaluation, and malicious behaviour detection. By leveraging machine learning algorithms, it is possible to analyze vast amounts of data generated by HPC systems and identify potential security threats in real-time. This can help to improve the usability and security of HPC systems by enabling early detection and mitigation of security incidents. As such, machine learning-based security solutions represent a promising approach towards enhancing the security of HPC systems in the future.

6. Conclusion

This paper provides an automated approach to analyze the login behaviour of users in HPC clusters via login attempts and IP addresses. The method aims to enable the prediction of unauthorized user activity when attempting to access the cluster. The method provided is highly robust and reliable, as it can generate output on highly imbalanced data and effectively address unsupervised problems. The model has demonstrated exceptional performance, achieving an F1-score of 0.98 and effectively detecting the majority of intrusions on the cluster. The probabilistic model is capable of detecting and capturing any activity that falls within the ambiguous or uncertain area (the "gray area").

References

[1] L. Alvarez, E. Ayguade and F. Mantovani, "Teaching HPC Systems and Parallel Programming with Small-Scale Clusters," 2018 IEEE/ACM Workshop on Education for High-Performance Computing (EduHPC), Dallas, TX, USA, 2018, pp. 1-10, doi: 10.1109/EduHPC.2018.00004.

[2] A. Ramirez, "Supercomputing: Past, present, and a possible future," 2011 International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation, Samos, Greece, 2011, pp. ii-ii, doi: 10.1109/SAMOS.2011.6045437.

[3] R. Bulusu, P. Jain, P. Pawar, M. Afzal and S. Wandhekar, "Addressing security aspects for HPC infrastructure," 2018 International Conference on Information and Computer Technologies (ICICT), DeKalb, IL, USA, 2018, pp. 27-30, doi: 10.1109/INFOCT.2018.8356835.

[4] S. Singh, K. R. Ramkumar and A. Kukkar, "Machine Learning Techniques and Implementation of Different ML Algorithms," 2021 2nd Global Conference for Advancement in Technology (GCAT), Bangalore, India, 2021, pp. 1-6, doi: 10.1109/GCAT52182.2021.9586806.

[5] R. Li, Q. Tao, Y. Luo and L. Yan, "Research on Practical Application of Data Analysis and Visualization," 2020 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), Zhangjiajie, China, 2020, pp. 78-81, doi: 10.1109/ICVRIS51417.2020.00026.

[6] I. Gupta, R. Gupta, A. K. Singh and R. Buyya, "MLPAM: A Machine Learning and Probabilistic Analysis Based Model for Preserving Security and

Privacy in Cloud Environment," in IEEE Systems Journal, vol. 15, no. 3, pp. 4248-4259, Sept. 2021, doi: 10.1109/JSYST.2020.3035666.

[7] Luo, Zhengping & Qu, Zhe & Nguyen, Tung & Zeng, Hui & Lu, Zhuo. (2019). Security of HPC Systems: From a Log-analyzing Perspective. ICST Transactions on Security and Safety. 6. 163134. 10.4108/eai.19-8-2019.163134.

[8] D. Liu, Y. Zhang, H. Zhang and F. Lou, "User behaviour control method for HPC system," 2021 China Automation Congress (CAC), Beijing, China, 2021, pp. 2918-2922, doi: 10.1109/CAC53003.2021.9727368.

[9] A. Prout et al., "Enhancing HPC security with a user-based firewall," 2016 IEEE High Performance Extreme Computing Conference (HPEC), Waltham, MA, USA, 2016, pp. 1-4, doi: 10.1109/HPEC.2016.7761641.

[10] Hou, Tao & Wang, Tao & Shen, Dakun & Lu, Zhuo & Liu, Yao. (2020). Autonomous Security Mechanisms for High-Performance Computing Systems: Review and Analysis. 10.1007/978-3-030-33432-1_6.

[11] Peisert, Sean. (2017). Security in high-performance computing environments. Communications of the ACM. 60. 72-80. 10.1145/3096742.

[12] https://nsmindia.in/

[13] H. He and E. A. Garcia, "Learning from Imbalanced Data," in IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263-1284, Sept. 2009, doi: 10.1109/TKDE.2008.239.

[14] A. Farid, N. Ali and M. u. Haq, "Reliability in Systems Based on Some Probability Models," 2018 International Conference on Applied and Engineering Mathematics (ICAEM), Taxila, Pakistan, 2018, pp. 120-125, doi: 10.1109/ICAEM.2018.8536309.

[15] M. Imran, A. M. Mahmood and A. Abdul Moiz Qyser, "An empirical experimental evaluation on imbalanced data sets with varied imbalance ratio," International Conference on Computing and Communication Technologies, Hyderabad, India, 2014, pp. 1-7, doi: 10.1109/ICCCT2.2014.7066742.

[16] S. J. Basha, S. R. Madala, K. Vivek, E. S. Kumar and T. Ammannamma, "A Review on Imbalanced Data Classification Techniques," 2022 International Conference on Advanced Computing Technologies and Applications (ICACTA), Coimbatore, India, 2022, pp. 1-6, doi: 10.1109/ICACTA54488.2022.9753392.

[17] Jang Bahadur Saini, D. . (2022). Pre-Processing Based Wavelets Neural Network for Removing Artifacts in EEG Data. Research Journal of Computer Systems and Engineering, 3(1), 43–47. Retrieved from https://technicaljournals.org/RJCSE/index.php/journal/article/view/40

[18] Mehta, E. S. ., & Padhi, S. . (2023). Quality and Defect Prediction in Plastic Injection Molding using Machine Learning Algorithms based Gating Systems and Its Mathematical Models. International Journal on Recent and Innovation Trends in Computing and Communication, 11(3s), 216–230. https://doi.org/10.17762/ijritcc.v11i3s.6183