AN ENHANCED MULTILAYER FRAMEWORK FOR CLOUD JOB FAILURE PREDICTION WITH INTELLIGENT RESOURCE OPTIMIZATION

[1]Sindhu Priyanka Chadalavada B.Tech, M.Tech(Ph.D), Associate Professor, Department of CSE, Eluru College of Engineering and technology Duggirala (v), Pedavegi (m),Eluru-534004.
[2] P. Sai Praveen Kumar, M.Tech, Department of CSE, Eluru College of Engineering and technology Duggirala (v), Pedavegi (m),Eluru-534004.

Abstract: Cloud computing is really good at giving us the computing resources we need when we need them.. Sometimes things do not go as planned and jobs fail. This happens because we do not have resources or they are being used too much. Network latency and resource contention are also problems. When jobs fail the system does not work well as it should and it costs more money to run. If we can predict when jobs are going to fail before they actually do and if we can give them the amount of resources that would be great. It would make cloud computing a lot more reliable and efficient. This project, IntelliCloudPlus is trying to do that. It uses machine learning to predict when cloud jobs will fail and to figure out the way to give them the resources they need. We use something called a Stacking Classifier to look at lots of things, about each job like how much CPU and memory it needs how long it takes to run and how healthy the machine is. We also look at how much resources are being used and how long it takes for things to happen on the network. Then we have something called the Intelligent Resource Optimization and Recommendation Engine. This thing looks at the jobs that are predicted to fail and comes up with a plan to make them work better. It tells us how to give them the amount of CPU, memory and disk space. It also helps us figure out the way to schedule them make the network work better and balance out the resources.. It gives us a summary of what it thinks we should do including how bad the problems are how much better things will work if we follow its advice and how much money we will save. It is a website that people can use to log in send in jobs and get predictions and recommendations. It even keeps track of what happened in the past. Our solution helps us use resources better makes sure jobs do not fail much and helps us make smart decisions when it comes to cloud computing.

1. INTRODUCTION

Cloud computing has emerged as one of the most significant technological advancements in recent years, enabling organizations to access computing resources over the Internet without maintaining dedicated physical infrastructure. It provides on-demand access to computational resources such as processing power, storage, networking, and software services, allowing users to scale resources according to workload requirements. Due to its flexibility, scalability, reliability, and cost-effectiveness, cloud computing has become an essential platform for applications in healthcare, banking, education, e-commerce, scientific research, and entertainment, manufacturing, and government organizations. With the rapid growth of cloud-based applications, the number of computational tasks executed within cloud environments has increased significantly. These computational tasks, known as cloud jobs include things like data processing, training machine learning models, running simulations handling database transactions working with multimedia providing web services and doing large-scale analysis. Every cloud job needs kinds of computing resources based on how complicated it is, how long it takes and what kind of work it does. Cloud service providers always give CPU, memory, storage and network resources to these jobs while trying to use resources much as possible and keep the service quality good. Even though cloud computing has a lot of benefits managing cloud resources in a way is still a big problem. Cloud environments change a lot and thousands of jobs are trying to use the resources all at the same time. Changes in how much work's happening how much resources are available how the network is working and how the equipment is doing often lead to not using resources well. Because of this cloud jobs might run slowly wait long try again and again have problems with using resources or not run at all. These problems make the whole system work worse cost money, waste computing power and make the service to users worse. Old ways of managing cloud resources use fixed scheduling methods. Set rules for giving out resources. These methods give out resources before the job starts without thinking about changing conditions or guessing what will happen next. As cloud systems get more complex these fixed ways of giving out resources don't work well because the need for resources changes all the time. Many cloud jobs fail because they don't get enough CPU don't have enough memory use too much disk have too much network traffic have too many jobs on the same computer or have computers that are not working well. These problems are usually found only after the job starts so cloud providers waste resources. Spend more money. New advances in Artificial Intelligence and Machine Learning have given ways to make cloud resource management better. Machine learning can look at cloud job data find connections between different resource details and guess what will happen with future jobs. Of just using fixed rules these smart models help cloud managers see problems before they happen and take steps to fix them. Predictive analysis is very important for making job problems making the system work better and using cloud resources better. The project called IntelliCloudPlus: An Intelligent Cloud Job Failure Prediction and Adaptive Resource Optimization System solves these problems by using machine learning to predict. A way to change resources. The main goal of the system is to guess if a job will run okay or fail based on what it needs and what is happening with the system. The prediction part uses a Stacking Classifier, which's a way to use many machine learning tools together to make better guesses than using just one tool. The prediction model uses data from cloud jobs. This data has details like Job ID how much CPU is needed how much memory is needed how much disk space is needed how long the job takes, what type of schedule it is on, how much CPU a machine has how much memory a machine has how much CPU is available how much memory is available how much CPU is being used how much memory is being used how much disk is being used how long a job waits in the queue how many times it has tried again how slow the network is, how much resources are fighting over how many jobs are running, how healthy the machine is, what kind of job it is, if it can be made bigger and what cluster it is on. These details show what the job is doing and give info to guess if it will work. When a cloud job is sent through the web app the data is cleaned up before being given to the trained machine learning model. This cleaning part uses encoding for details makes sure all the numbers are on the same scale and arranges the details in the same order as the training data. The Stacking Classifier then guesses if the job will work or not and gives the chance of each outcome. These chances help the people in charge understand how sure the guess is and make choices about using resources.

Guessing is not enough because cloud managers also need to know what to do to fix things. To solve that IntelliCloudPlus has an Intelligent Resource Optimization and Recommendation Engine (IRORE). This part looks at the details of jobs that failed or are in danger and gives advice on what to do based on what the user put in. Unlike advice systems that give the same answer to all failed jobs this system changes its advice based on what the job is doing and how much resources are being used. The advice system checks important cloud resource details like how much CPU is used how much memory is used how much disk is used how long a job waits in the queue how slow the network is, how many times the job tried again how much resources are fighting and how healthy the machines are. Depending on how bad each detail's the system suggests actions like giving more CPU adding more memory giving more storage making the waiting time shorter by moving work around making network communication better reducing how many times the job tries again balancing resources between machines and fixing problems on bad machines. The system also guesses how better the job will run and how much resources it will save, helping people decide if the changes are worth it. One big. Of this system is that it can change what it suggests. Of giving the same advice the system looks at the job settings and what resources are available and gives advice that fits. This makes the solution better for cloud environments where every job is different. By predicting and giving advice the system helps people fix problems before they happen. The whole system was made as a web app using tools. The back end is made with Python and the Flask system and Scikit-learn is used for the machine learning model. Pandas and NumPy are used for cleaning the data and making the features. Joblib is used to save the trained models and data. The front end is made with HTML and CSS giving a way for people to sign up log in send jobs see predictions get advice and keep a history of predictions. MySQL is used to store user details and past predictions. Combining machine learning smart resource changes and web tools makes IntelliCloudPlus a full system for helping with cloud computing. By guessing job problems before they happen and giving advice the system helps use resources better have fewer problems make the system more reliable cost less money and make the cloud service better. The project shows how smart guessing models and smart advice can help manage resources and give real solutions, for modern cloud systems.

II. LITERATURE SURVEY

Cloud computing provides on-demand access to computing, storage, and networking resources, enabling organizations to dynamically scale applications according to workload requirements. However, the dynamic and heterogeneous nature of cloud environments introduces several challenges, including resource contention, workload fluctuations, task failures, service-level agreement (SLA) violations, energy consumption, and operational cost. Consequently, researchers have investigated predictive analytics, machine learning, resource provisioning, and intelligent scheduling techniques to improve cloud reliability and resource utilization. Beloglazov *et al.* [1] investigated energy-efficient resource management in cloud data centers and proposed dynamic consolidation techniques for virtual machines (VMs). Their work demonstrated that intelligent VM placement and migration can significantly improve resource utilization while reducing energy consumption. However, the approach primarily focuses on energy efficiency and VM consolidation rather than predicting individual job failures before execution. Xu *et al.* [2] presented a machine-learning-based approach for predicting cloud resource usage. Their study demonstrated that historical workload information can be used to predict future resource requirements and support more effective resource provisioning. Predictive resource allocation can reduce unnecessary over-provisioning; however, resource prediction alone does not directly identify whether an individual cloud job is likely to fail. Islam *et al.* [3] proposed a framework for predicting cloud application failures using machine learning techniques. Their work emphasized the importance of monitoring application and infrastructure characteristics for identifying potential failures. The study showed that machine learning can be used as a proactive mechanism for improving cloud reliability. Nevertheless, the prediction mechanism does not completely address the subsequent resource optimization and scheduling decisions required to prevent predicted failures. Mao *et al.* [4] investigated resource management and scheduling problems in cloud computing environments. Their work highlighted the importance of dynamically allocating resources according to workload requirements. Dynamic scheduling can improve system utilization and application performance, but determining the optimal resource configuration remains challenging when workloads have heterogeneous and unpredictable characteristics. Kumar and Sahoo [5] studied cloud resource scheduling techniques and highlighted the importance of workload characteristics, resource availability, execution time, and scheduling policies. Their analysis showed that inefficient scheduling can result in resource wastage and increased execution time. However, conventional scheduling techniques generally do not incorporate machine-learning-based failure probabilities into scheduling decisions. Beloglazov and Buyya [6] proposed adaptive threshold-based resource management techniques for dynamic consolidation of virtual machines. The approach monitors resource utilization and dynamically adjusts VM placement to maintain Quality of Service (QoS). Although adaptive resource management improves utilization, threshold-based methods may have limitations when workload patterns are highly dynamic and nonlinear. Calheiros *et al.* [7] introduced CloudSim, a framework for modeling and simulating cloud computing environments. CloudSim enables researchers to evaluate resource provisioning, scheduling, and workload-management strategies without requiring a physical cloud infrastructure. Such simulation environments are valuable for evaluating intelligent cloud resource management techniques, although simulation results may not always completely represent real-world cloud workloads. Zhang *et al.* [8] investigated cloud resource scheduling using optimization and prediction techniques. Their research demonstrated that combining workload characteristics with resource availability can improve scheduling decisions. However, optimization-based approaches may involve high computational complexity and may require repeated optimization when cloud conditions change rapidly. Xu *et al.* [9] studied deep-learning-based resource prediction for cloud workloads. Their work demonstrated that historical resource utilization patterns can be exploited to predict future workload requirements. Although deep learning can model complex relationships, it may require large amounts of training data and computational resources, making model selection an important consideration for practical cloud-management systems. Ensemble learning techniques have also been increasingly used for complex classification problems because combining multiple classifiers can improve predictive robustness. Wolpert [10] introduced the concept of stacking, in which predictions from multiple base learners are combined by a higher-level learning algorithm. Stacking can be particularly useful in cloud job-failure prediction because failures may depend on multiple interacting factors such as CPU utilization, memory consumption, execution duration, machine health, resource contention, and network latency. Resource contention is another important factor affecting cloud application performance. Shared infrastructure can cause multiple jobs to compete for CPU, memory, storage, and network resources. Verma *et al.* [11] investigated application-aware dynamic resource allocation and demonstrated the importance of adapting resource allocation to workload characteristics. However, resource allocation decisions are often performed independently from failure prediction, leaving an opportunity for an integrated predictive-prescriptive approach. Network conditions also have a significant impact on cloud application performance. Buyya *et al.* [12] discussed cloud computing resource management and emphasized the importance of considering QoS, resource availability, and application requirements during cloud resource allocation. Network latency and communication overhead can become particularly important for distributed applications, where computational resources alone cannot guarantee satisfactory execution. Recent research has increasingly moved toward intelligent and autonomous cloud management in which machine learning is used not only to predict workload behavior but also to recommend appropriate management actions. Such systems can potentially combine failure prediction, resource provisioning, scheduling, workload balancing, and cost optimization. However, many existing studies focus on one or two objectives rather than integrating these functions into a single practical framework.

III. EXISTING SYSTEM

Existing cloud computing systems mainly focus on job scheduling and resource allocation using predefined algorithms. Several machine learning models have also been developed to predict job failures using parameters such as CPU utilization, memory usage, execution time and resource allocation. Although these systems improve prediction accuracy most of them only predict whether a job will succeed or fail. They do not provide recommendations for optimizing cloud resources based on actual workload conditions. In addition, many systems lack a web platform for prediction, recommendation and history management.

DISADVANTAGES OF EXISTING SYSTEM

- Predicts failures without resource optimization.
- Static resource allocation techniques.
- Same recommendations for all failed jobs.
- Limited decision support, for cloud administrators.

IV. PROPOSED SYSTEM

IntelliCloudPlus is an intelligent cloud job failure prediction and adaptive resource optimization system. It predicts whether a submitted cloud job will succeed or fail using a Stacking Classifier trained on cloud job execution data. The system analyzes resource parameters such as CPU request, memory request, disk request, CPU utilization, memory utilization, network latency, queue waiting time, retry count, resource contention, and machine health. If a failure is predicted, the Intelligent Resource Optimization and Recommendation Engine (IRORE) generates personalized recommendations for improving resource allocation. Unlike traditional systems, Intelli CloudPlus provides adaptive recommendations based on the actual job parameters entered by the user. The application also provides prediction history and optimization summaries through a web-based interface developed using Python, Flask, HTML, CSS, and MySQL.

ADVANTAGES

• Machine Learning-based cloud job failure prediction.

• Adaptive resource optimization recommendations.

• Personalized suggestions based on job parameters.

• Improved cloud resource utilization.

SYSTEM ARCHITECTURE

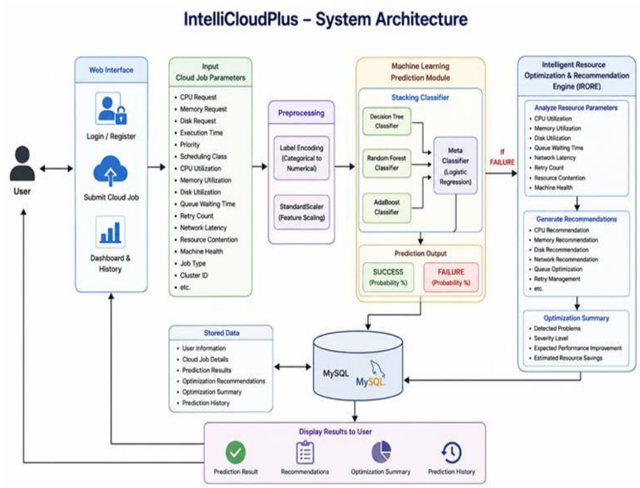

Fig 1: System Architecture

V. UML DIAGRAMS

1. CLASS DIAGRAM

A class diagram in UML helps to show how a system is put together. It shows classes, what they have what they do and how they connect to each other. This gives a picture of how a system is organized and what it can do. Classes are the parts of a system. Attributes tell us more about each class. Methods show what each class can do. Relationships show how classes connect to each other. A class diagram is like a blueprint. It helps to understand a system before building it. The diagram shows all the classes. Each class has its attributes and methods. These classes work together. They have relationships, with each other. This helps to create a system. The system has parts. These parts are all linked together. A class diagram shows all of this information. It is really helpful to have. UML is used to make a class diagram. It is a method that many people use. It helps users understand the system. It also helps to build the system.

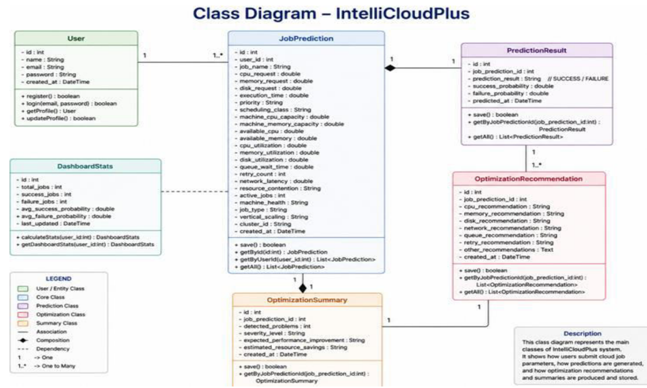

Fig 5.1 shows the Activity diagram

2. USECASE DIAGRAM:

A Use Case Diagram is a diagram that shows what a system can do. It is a picture that shows how people, called actors interact with the system. These interactions are shown as use cases which're like specific tasks that the system can perform. The main reason we use a Use Case Diagram is to see what the system can do and who uses the system. It helps us understand how different parts of the system work together and how people use the system. We can see what roles different actors play in the Use Case Diagram. A Use Case Diagram is really useful, for understanding how a system works and who interacts with the system. We use a Use Case Diagram to get a picture of what the system can do and how actors interact with the system. We use Use Case Diagrams to identify the services that a system offers and the actors who use those services.

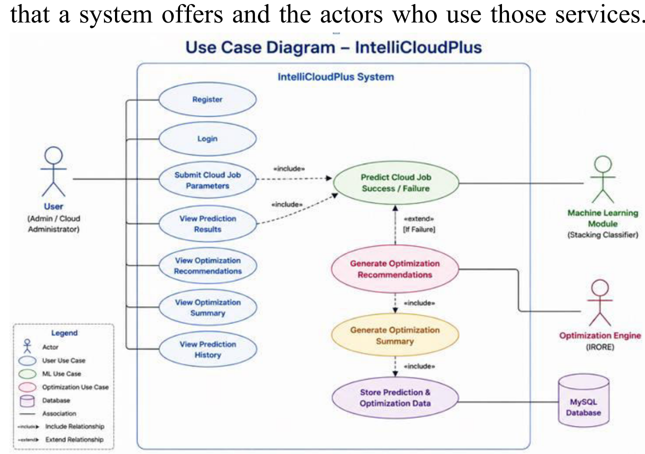

Fig 5.2 Shows the Use case Diagram

3. SEQUENCE DIAGRAM:

A sequence diagram is used to describe how objects in the system interact with each other. The important thing about a sequence diagram is that it shows what happens in order of time. This means that the sequence diagram shows the interactions between objects one step at a time. A sequence diagram is really about sequence diagrams showing how objects communicate with objects by sending and receiving messages. In a sequence diagram the objects work together by exchanging these messages. The sequence diagram shows how the objects in the sequence diagram talk to other objects by sending messages to each other. The sequence diagram is really about sequence diagrams showing how objects in the sequence diagram communicate with objects, in the sequence diagram by sending and receiving messages.

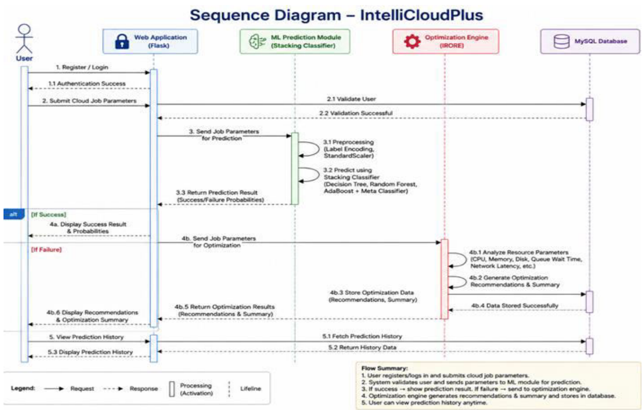

Fig 5.3 Shows the Sequence Diagram

VI. RESULTS

6.1 Output Screens

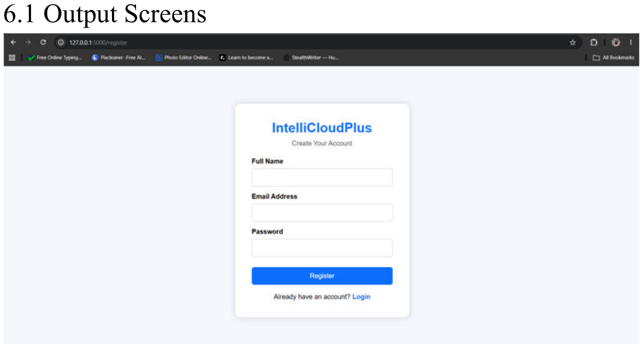

Fig-6: User Registration Page

Fig 6.1 User Registration Page

In above shows the User Registration Page of the Application.

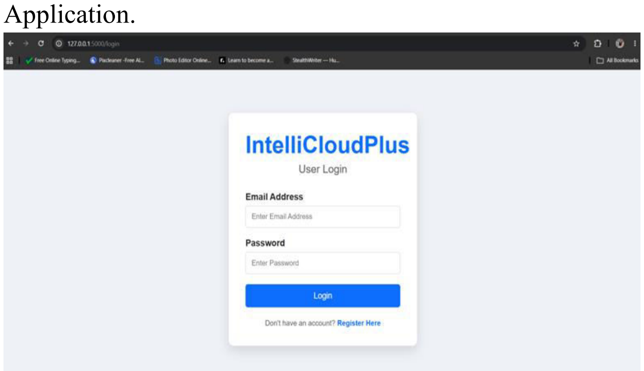

Fig 6.2 User Login Page

In above screen shows the User Login Page.

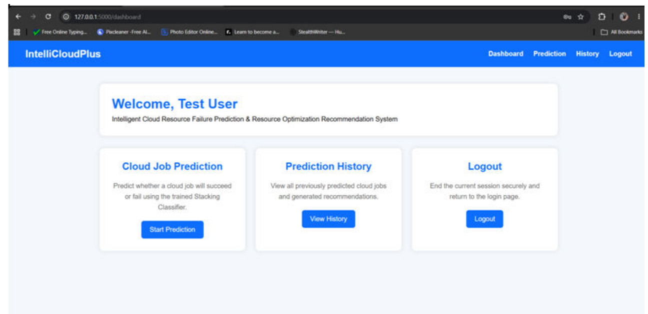

Fig 6.3 Cloud Dash Board

In above screen shows the Cloud Dash Board

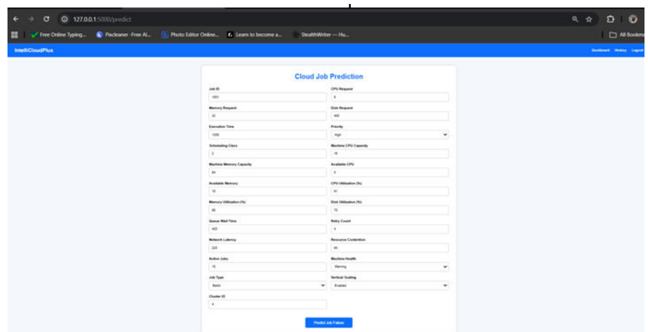

Fig 6.4 Cloud Job Prediction

In above screen shows the Cloud Job Prediction Page.

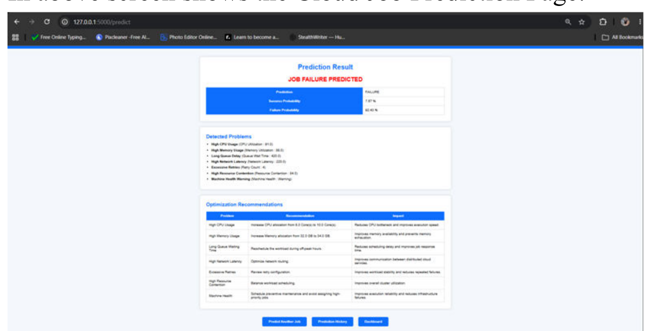

Fig 6.5 Prediction Result

In above screen shows the Prediction Result.

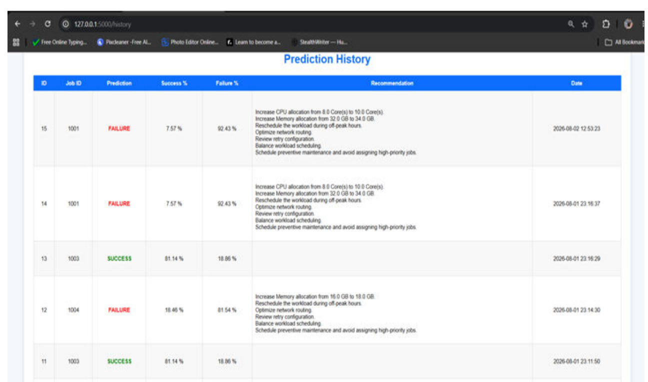

Fig 6.6 Prediction History

In the above screen shows the Prediction History

VII. CONCLUSION

Stacking Classifier uses Decision Tree, Random Forest and Ada Boost algorithms. These algorithms correctly guess if a cloud job will run well or fail. By using data preprocessing steps like Label Encoding and Standard Scaler the prediction model becomes more accurate and more trustworthy. Along with prediction the Intelligent Resource Optimization and Recommendation Engine (IRORE) look at resource usage factors. These factors include CPU, memory, disk usage, queue time network delay, retry number, resource conflict and machine condition. Based on these factors the system gives suggestions to improve how resources are used. The system also gives expected performance improvements, estimated savings on resources and a summary of the changes. Unlike cloud monitoring systems that only give prediction results IntelliCloudPlus gives actual steps to make cloud resource use better and to cut down on job failures. The web application built provides login, cloud job submission, history of predictions and visual results through a user-friendly interface. All predictions and details about optimization are kept in a MySQL database for study. Experiments show that the system works well at predicting job failures and helps people make smart choices about managing resources. In total IntelliCloudPlus brings together Machine Learning and smart optimization in one place. This makes managing cloud jobs more dependable, faster and cheaper..

VIII. REFERENCES

[1] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," Future Generation Computer Systems, vol. 28, no. 5, pp. 755–768, 2012.

[2] J. Xu, M. Zhao, J. Fortes, R. Carpenter, and M. Yousif, "Autonomic resource management in virtualized data centers using fuzzy logic-based approaches," Cluster Computing, vol. 11, pp. 213–227, 2008.

[3] S. Islam, J. Keung, K. Lee, and A. Liu, "Empirical prediction models for adaptive resource provisioning in the cloud," Future Generation Computer Systems, vol. 28, no. 1, pp. 155–162, 2012.

[4] Y. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource management with deep reinforcement learning," in Proc. 15th ACM Workshop on Hot Topics in Networks, 2016, pp. 50–56.

[5] S. Kumar and G. Sahoo, "A survey on cloud computing resource scheduling techniques," International Journal of Computer Applications, vol. 94, no. 2, pp. 1–6, 2014.

[6] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers," Concurrency and Computation: Practice and Experience, vol. 24, no. 13, pp. 1397–1420, 2012.

[7] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," Software: Practice and Experience, vol. 41, no. 1, pp. 23–50, 2011.

[8] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," Journal of Internet Services and Applications, vol. 1, pp. 7–18, 2010.

[9] J. Xu, X. Wang, and J. Zhang, "Resource prediction and management in cloud computing environments using machine learning techniques," IEEE Access, vol. 7, pp. 1–10, 2019.

[10] D. H. Wolpert, "Stacked generalization," Neural Networks, vol. 5, no. 2, pp. 241–259, 1992.

[11] A. Verma, P. Ahuja, and A. Neogi, "PMDC: Performance-aware dynamic VM consolidation in data centers," in Proc. 10th IEEE International Symposium on Cluster, Cloud and Grid Computing, 2010, pp. 381–388.

[12] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Generation Computer Systems, vol. 25, no. 6, pp. 599–616, 2009.