National College of Ireland

Improving fault tolerance of a task in cloud using ensemble approach

MSc Research Project
Cloud Computing

Divesh Soneji
Student ID: X21172749

School of Computing
National College of Ireland

Supervisor: Prof. Yasantha Samarawickrama

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Divesh Soneji
Student ID:	X21172749
Programme:	Cloud Computing
Year:	2022-2023
Module:	MSc Research Project
Supervisor:	Prof. Yasantha Samarawickrama
Submission Due Date:	14/08/2023
Project Title:	Improving fault tolerance of a task in cloud using ensemble approach
Word Count:	8817
Page Count:	23

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	13th August 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Improving fault tolerance of a task in cloud using ensemble approach

Divesh Soneji
X21172749

Abstract

The presented study investigates the improvement of fault tolerance in cloud tasks through the utilisation of an ensemble approach. The study conducted provides a comparative analysis of different models, with a specific emphasis on the classification of tasks into two categories: success and failure. The study involved a comparison between an ensemble model and two machine learning models, with the evaluation of various metrics including F1 score, accuracy, and recall. The performance of the models was evaluated using a split of 80 percent training data and 20 percent testing data. Three case studies were conducted, which involved the utilisation of the K-Nearest Neighbours (KNN) model, Artificial Neural Network (ANN) model, and an ensemble model. The models are evaluated based on the accuracy, root means square error (RMSE), R-Square , F1-score and Recall . Out of the three models considered, the ensemble model that combines both the KNN and ANN using logistic regression demonstrates 15 percent improvement from KNN and 1 percent improvement from ANN in accuracy as seen in the performance metrics. Despite its efficacy, this approach presents challenges in terms of heightened resource demands and increased complexity. The results emphasise the significance of effectively implementing an ensemble model in task scheduling algorithms in data centres, as it facilitates a robust approach to ensure fault tolerance.

1 Introduction

The cloud has emerged as a crucial element within the technological landscape and has proven its capacity to facilitate the growth of various sectors. Cloud computing involves the examination and enhancement of algorithms to optimise the efficiency of different aspects, including resource allocation, load balancing, and reliability. Additionally, it aims to improve the quality of service by reducing the average time between system failures. Currently, there is a shift taking place towards the implementation of an automated approach, which aims to minimise instances of human error and eliminate redundant tasks. The automation of decision-making components within the cloud is being achieved through the utilisation of Machine Learning (ML) and Deep Learning (DL) techniques. The emergence of this phenomenon has given rise to a novel area of study referred to as intelligent cloud computing. This enables us to focus on improving the cloud infrastructure through the implementation of diverse intelligent methodologies. Enhancing fault tolerance, a fundamental aspect of cloud computing, has been a focal point of scholarly investigation.

There are various categories of failure in cloud which can create a cascading event of failures. To mitigate these failure the existing system use various measure to ensure the continuity of service in the event of a bottleneck. The implementation of fault management in a cloud environment serves to augment the system's resilience and establish a sense of dependability for customers in their utilisation of cloud service providers, thereby ensuring fault tolerance. There exist three discrete classifications of fault tolerance techniques. The Reactive method is widely utilised by cloud providers as the primary approach for swiftly allocating resources in reaction to service failures. The utilisation of this particular methodology is considered suitable for attaining fault tolerance, despite the fact that it involves a substantial amount of additional resources. The length of time for overhead can range from a minimum of 5 seconds to a maximum of 15 seconds, which has the potential to disrupt the operational activities of the customer's business. The term "overhead" pertains to the temporal requirement for the initiation of a new resource or the rebooting of a service. The reactive approach encompasses various techniques, such as Checkpointing/Restart, Replication, Task Resubmission, Retry, and others.

Furthermore, there are concerns surrounding repetitive tasks, specifically the process of determining the threshold value. This procedure is performed manually and therefore carries the risk of human error. There is an increasing inclination to adopt automated systems with self-awareness capabilities, specifically in the domain of failure comprehension, as a means to reduce the occurrence of repetitive tasks. Through the utilisation of failure logs monitoring, it becomes feasible to anticipate the likelihood of failure occurrences and subsequently initiate a resource allocation process using the aforementioned reactive methodologies. The utilisation of proactive measures for the purpose of fault detection is widely known as Proactive Methods. There are various methodologies that have been developed, such as Self-healing, Pre-emptive Migration, Monitoring, S-Guard, and Software Rejuvenation. Extensive research has been undertaken regarding the phenomenon of self-healing and the acquisition of preemptive migration using machine learning methodologies.

Numerous studies have utilised diverse machine learning algorithms to predict the most suitable threshold value for preemptive migrations, as well as to monitor resources through the use of heartbeats. By engaging in the process of predicting reliability, we can make informed conclusions about the threshold quantity. Currently, there exists a novel field of inquiry that combines the reactive and proactive methodologies, commonly known as the resilient approach. The methodology has undergone refinement in order to effectively address failures, utilising a pre-existing dataset of failures as a basis. This study aims to develop a framework that incorporates the capability to forecast Byzantine failure through the prediction of task failure in cloud computing. This can be achieved by identifying features such as the requested resource and the state of the task event. The model can subsequently be trained to acquire knowledge of the pattern of task failure through the utilisation of a machine learning algorithm. In this study, three distinct models will be employed for the classification task: K-nearest Neighbour (KNN), Artificial Neural Network (ANN), and an Ensemble model that combines both KNN and ANN using the stacking method. The Stack method incorporates a Meta-Learner that combines the outcomes using logistic regression. In subsequent applications, this model can be employed to detect instances of task failure during task scheduling by assigning task priority and subsequently directing it to a distinct queue. One advantage of this approach is that the allocation of resources for the task will not occur until the scheduler has processed the queue. This approach has the potential to decrease resource

allocation, resulting in cost reduction for cloud services and facilitating the adoption of green computing practises. Hence, the focus of our research will revolve around the subsequent inquiries.

- What is the efficacy of an ensemble approach utilising deep learning models jointly for the purpose of forecasting task failure in cloud computing environment?
- How effective are various machine learning models compared to an Ensemble model for enhancing predictive accuracy for failure of task in cloud environment ?

Outline: This document comprises a comprehensive literature review discussing the diverse range of techniques employed by previous researchers to enhance fault tolerance. Following this, a methodology is presented, outlining our proposed approach. Subsequently, an implementation section details the specific techniques employed to realise the proposed approach. Finally, an evaluation section is included, which delves into the experiments conducted to assess the effectiveness of the proposed approach. The Discussion section will serve as a platform for analysing and interpreting the results obtained from the experiment. In this section, the implications and significance of the findings will be thoroughly examined and discussed. On the other hand, the Conclusion section will provide a comprehensive overview of the research, encompassing a synthesis of the key findings and their broader implications. Additionally, this section will outline potential avenues for future research and exploration i .

2 Related Work

2.1 Survey Papers

The anticipation of faults has been a central focus of study throughout the years. Different models have been employed to enhance problem identification in cloud computing and mitigate faults through proactive and reactive approaches. The primary objective of this project is to employ machine learning techniques in order to detect occurrences of cloud malfunction. The topic at hand has undergone thorough investigation; nevertheless, there exist certain areas within the research that necessitate additional exploration. According to a recent scholarly article, various challenges associated with cloud computing can be addressed through the implementation of corrective strategies such as check pointing, proactive migration, retrying, rescheduling tasks, and software rejuvenation. Both the reactive and proactive approaches advocate for the implementation of all these mitigation strategies. This survey paper provides a thorough analysis of the different shortcomings that can arise in cloud computing, as well as the prevailing solution currently employed by most cloud service providers and data centres. The article also cites recent studies on a Resilient approach that utilises artificial intelligence and machine learning to forecast cloud resource failures with a high likelihood. Multiple models have been employed for the prediction of failures, encompassing failures in both software and hardware domains. The survey paper serves as a tool for categorising problems and providing valuable analysis on the potential ramifications of system malfunction, such as the potential for initiating a cascade of events within the cloud infrastructure. By directing attention towards particular concerns such as application failure, it becomes feasible to enhance the quality of service (QoS) Shahid et al. (2021).

The findings of a recent survey focused on fault tolerance have shed light on the challenges faced within distributed systems, specifically in the context of cloud computing. The aforementioned statement underscores the potential difficulties that may arise from the existence of various cloud components that engage in mutual interactions within a given system. The factors mentioned above include the need for a comprehensive Quality of Service (QoS), the ability to withstand faults, diligent monitoring, understanding power consumption, distinguishing between cloud native and cloud enabled, automation, and a lack of established frameworks. The task of prioritising components is of utmost importance for system architects, as each component possesses distinct characteristics and responsibilities. The present study subsequently investigated the effective implementation of Machine Learning (ML) in a comprehensive and reliable manner. The application of machine learning and artificial intelligence has demonstrated its efficacy in forecasting cloud malfunctions. The dataset was cited and the potential of machine learning to enhance research on self-aware fault tolerance was deliberated Agarwal and Kotakula (2022).

2.2 Prediction of hardware failure using SMART attributes

In an independent study, an Artificial Neural Network (ANN) was employed to predict the potential failure of a hard disc in a corresponding server. The combination of ANN and Self-Monitoring, Analysis, and Reporting Technology (SMART) characteristics has been observed to improve the precision of hard disc failure prediction. Consequently, the fault tolerance management module is empowered to take proactive measures in order to prevent the allocation of virtual machines to servers that demonstrate a potential vulnerability to failure. This study provides a comparative analysis of the ANN and KNN algorithms, highlighting their respective performance in terms of accuracy, which is reported to be 90 percent. The practical implementation of cloud computing architecture entails the utilisation of a central controller, which serves as the primary entity responsible for receiving user requests and subsequently allocating them to physical machines. The second function is concerned with the synchronisation of multiple modules that are responsible for facilitating the efficient management of the cloud infrastructure. The architectural design being proposed incorporates databases that are administered by Hadoop and MapReduce modules, alongside a secondary controller that oversees the system's status and provides notifications in the event of any modifications. This article presents a methodology that comprises four essential stages. These stages include data acquisition from diverse sources, data preparation through the removal of anomalous data, the development of a precise predictive model, and the integration of the established model into a comprehensive load forecasting framework. ANN is utilised for the purpose of discerning and classifying data that exhibits either absence or aberration. The study conducted an analysis of the data collected by Backblaze, a cloud storage provider, over the time frame from December 2015 to December 2018. Machine learning algorithms were utilised to predict the occurrence of failure based on the values of the SMART attributes Ragmani et al. (2020).

2.3 Prediction of Task Failure in Distributed Systems

Subsequently, another researcher introduced a methodology aimed at forecasting task failure by employing diverse approaches. The author has employed five machine learn-

ing algorithms and evaluated their respective performances by assessing their accuracy. The author has endeavoured to address the issue of classification by employing three distinct categories of algorithms, specifically regression, overtime, and ensemble. The logistic regression (LR) algorithm has been widely studied and recognised as a highly researched regression technique in statistical analysis. The Decision Tree (DT) algorithm is frequently employed as the principal machine learning technique for classification tasks, while Random Forest (RF) is classified as an ensemble algorithm. The scikit-learn library is commonly employed for the implementation of machine learning algorithms, except for XGBoost, which is executed using the XGBoost library. The maximum number of indices in the logistic regression model has been adjusted, but the default value is insufficient for ensuring convergence across all solvers. The Solver algorithm is employed as an optimisation technique for calculating the loss of the Linear Regression model. The author has additionally incorporated three discrete variations of the LSTM and DL models, each distinguished by a distinct number of layers. The deep learning model consists of three distinct sub-models: the Single Layer Long Short-Term Memory (LSTM) with three layers, the Bi-Layer LSTM with two hidden layers, and the Tri-LSTM with three hidden layers. To achieve this objective, the algorithm incorporates a dense layer to ensure the generation of a singular value for prediction. Moreover, the training procedure is halted if there is no improvement in the validation loss metric after 10 epochs. The conclusion of the study indicated that XGBoost exhibited superior accuracy in classification compared to other models, whereas Random Forest (RF) and Decision Tree (DT) models were found to be more suitable for task-level predictionTengku Asmawi et al. (2022).

A previous study has demonstrated the implementation of a scheduling scheme for energy-conscious fault tolerance. This scheme utilises deep neural networks to predict failures and schedule tasks within a replica for execution. During the initial phase, the task is subjected to testing in order to assess its likelihood of encountering failure. Consequently, it is classified as either prone to failure or not prone to failure. The objective has been re-framed from duplication to more complex tasks by employing the vector reconstruction methodology. Following this, a virtual machine is assigned the responsibility of executing these tasks. The utilisation of this particular approach leads to a reduction in energy consumption, consequently guaranteeing the preservation of Quality of Service (QoS). Furthermore, the utilisation of gradient descent by the author serves as a method for reducing prediction error within the framework of failure. The author has examined the influence of various resources on the occurrence of task failure. In contrast to the previously mentioned study, which provides a comprehensive examination of five traditional machine learning models and three deep learning models, the evaluation of energy consumption related to the scheduler for task rescheduling was not included. The main discovery of the research paper concerns the utilisation of vector bin for the purpose of rescheduling the super task on an appropriate host. This approach utilises predictive techniques to optimise the scheduling algorithm by assigning accurate numerical values to the parameters of the algorithm Marahatta et al. (2021).

Another scholarly article focuses on the prediction of failure using Long Short-Term Memory (LSTM) technology. However, it is important to note that LSTM is not capable of effectively handling multiple inputs. Therefore, this study provides a comprehensive examination of Bi-LSTM, alternatively known as Bidirectional Long Short-Term Memory, which integrates a greater quantity of input characteristics. The objective of this study is to ascertain the essential characteristics that need to be considered in the development of our prospective model. Both the training and testing of the model will incorporate the

Google cluster trace dataset. The findings indicate that the algorithm produces output data in both the forward and backward directions in order to adjust the weights of input features that are either close or distant. Furthermore, the assessment is conducted by comparing it to other exemplary models, taking into consideration their accuracy, F1 score, precision, and recall. Nevertheless, the accuracy of forecasting may be reduced when temporal intervals exceed a specific threshold. The current outcome is a result of the meticulous evaluation of the trade-off between the magnitude of the temporal interval and the precision of the forecast. The findings of the research indicate that Bi-LSTM demonstrated a predictive accuracy of 90 percent when the minimum temporal interval was set at 15 minutes and the size requirements were met Gao et al. (2022)

Another academic article that introduces a task scheduling framework that integrates failure awareness, allowing for the real-time prediction of task termination status and the subsequent execution of suitable remedial actions. The existence of this characteristic has led to a substantial portion of customers transferring their application tasks to cloud-based platforms. The framework demonstrates a notable capacity to safeguard around 40 percent of tasks that are projected to encounter failure, as evidenced by the Alibaba dataset, by effectively executing corrective measures. As a consequence, the conservation of cluster resources such as central processing units (CPUs) and random access memory (RAM) is achieved. Furthermore, the problem of action selection is formalised in this study through the utilisation of an Integer Linear Programming (ILP) model. Additionally, it presents a heuristic optimisation method with the objective of reducing the likelihood of task failure and resource utilisation. In the realm of cloud computing, it is not uncommon for tasks to experience failures as a result of various factors, including but not limited to software defects, hardware malfunctions, and inadequate resource allocation. The presence of such failures can potentially exert a detrimental influence on the Quality of Service (QoS) being provided. The existing corpus of academic literature primarily focuses on the anticipation of task failure, while offering limited consideration to subsequent remedial measures following failure Alahmad et al. (2021).

The primary objective of this study is to acquire a thorough comprehension of the characteristics associated with job failures, with the intention of improving the dependability of cloud infrastructure. This investigation specifically focuses on the viewpoint of cloud service providers. The reliability of cloud applications can be influenced by various factors, including the characteristics of the task, the configurations of the cloud, and the dynamic states of the cloud system. This research conducts a statistical analysis of job and task failures in order to identify potential associations between these failures and important scheduling constraints, node operations, and user attributes in the context of cloud computing. This study provides a comprehensive analysis of cloud failures, focusing on four key dimensions. The application process is influenced by various factors, such as the availability of different programs, the number of tasks associated with a specific job, and the individuals responsible for supervising the job. Cloud factors consist of various components, including node failures and maintenance operations. The configurations include scheduling constraints and policies regarding the maximum allowable number of re-submissions for a failed task. The present state of real-time execution primarily revolves around the utilisation of central processing unit (CPU) and memory resources during runtime. The researchers of this study put forth a number of potential strategies for enhancing the dependability of cloud applications, as suggested by their empirical observations. The strategies encompass proactive maintenance of nodes and the implementation of limitations on the frequency of job re-submissions. There exists a significant

level of interest in understanding the impacts of job scheduling and node maintenance on occurrences of job failures Jassas and Mahmoud (2021).

Throughout history, traditional methodologies have often focused on the application of rule-based systems and heuristic techniques. However, the adoption of sophisticated machine learning algorithms has led to a notable shift in methodology, with a growing inclination to leverage these techniques for predictive purposes within the realm of cloud computing. Multi-Layer Perceptrons (MLPs), a type of artificial neural network, have demonstrated considerable promise in various prediction tasks. The ability to accurately depict complex non-linear connections makes them well-suited for integration into cloud environments, which inherently exhibit dynamism and heterogeneity. While multi-layer perceptrons (MLPs) have found applications in diverse domains, their application in predicting job failures in cloud computing is a relatively recent development. Hyper-parameter tuning is a crucial aspect of training robust machine learning models, as it enhances the predictive capabilities of multi-layer perceptrons (MLPs). A multitude of academic studies have emphasised the importance of feature engineering in enhancing the effectiveness of predictive models. In the domain of cloud systems, the incorporation of specific characteristics, such as system logs, metrics concerning resource utilisation, and historical data regarding job failures, plays a vital role in determining the effectiveness of predictive models. In conclusion, the current body of scholarly work underscores the potential of machine learning methodologies, particularly neural networks like Multi-Layer Perceptrons (MLPs), in effectively addressing the challenges related to forecasting job failures in cloud systems. The growing intricacy of cloud environments is expected to amplify the importance of advanced machine learning models Vani and Sujatha (2022).

A recent study has conducted an analysis of comprehensive workload traces, specifically focusing on those that have been publicly disclosed by Google. The studies mentioned above have indicated that a significant portion of time within the cluster was allocated to the execution of tasks that ultimately did not achieve successful completion. The aforementioned findings highlight the crucial need for a thorough understanding of the mechanisms and justifications underlying job termination in large-scale systems. The exponential growth of data volume in these systems has not been accompanied by a proportional improvement in reliability and dependability. The matter of reliability transcends the confines of an individual system. The problem of job reliability poses a significant challenge in both conventional high-performance computing (HPC) systems, which are prone to frequent application aborts, and cloud computing environments that execute diverse workloads on complex software stacks and heterogeneous hardware. In both contexts, occupations have exhibited an increased susceptibility to faults and errors. The extant literature offers valuable insights into the specific characteristics that contribute to unsuccessful executions within systems such as Google's cluster. Nevertheless, there is still a dearth of understanding regarding the generalized ability of these findings to different computing clusters or various types of large-scale systems. Furthermore, there is a scarcity of scholarly literature that provides a comprehensive understanding of the practical implications of these studies in relation to improving unsuccessful job executions or enhancing clusters through alternative means El-Sayed et al. (2017).

The primary focus of all the researchers work revolves around the proactive strategy for improving fault tolerance. Furthermore, the majority of the article focuses on conventional machine learning models, although certain models, such as Bi-LSTM, may incorporate modifications in their logic. However, it is important to note that these models are prone to over-fitting and may exhibit sub-optimal performance. Another as-

pect that must be taken into account is the pre-processing of the data. It is important to recognise that not all data can be utilised for the development of a failure prediction model. Therefore, it is necessary to thoroughly examine the purpose of the dataset and make appropriate modifications and manipulations to extract relevant information. Subsequently, the dataset must undergo pre-processing in order to facilitate the model's comprehension and generate a set of features that will enhance the accuracy of the model. Upon recognising a recurring pattern of issues in the aforementioned article, our attention will now be directed towards addressing some of the underlying concerns. In order to address certain challenges, it is imperative to inquire whether there exists an optimal methodology for forecasting task failure, and if so, how one should determine the selection of such an approach ? Furthermore, what are the advantages of selecting a complex model, such as an ensemble model, for the purpose of predicting task failure? What is the comparative performance of the ensemble model in relation to other machine learning models for task failure in cloud ? Lastly, it is important to consider the necessary feature engineering techniques when working with the Google Cluster Trace dataset. In the subsequent research, we will engage in a concise examination of the aforementioned inquiries.

3 Methodology

In our study, the Google trace dataset has been taken into consideration. The dataset comprises information obtained from Google Borg servers, encompassing a total of eight distinct borg cells. The system furnishes data regarding CPU utilisation, requested CPU utilisation, and memory allocation for each job. Additionally, it provides information pertaining to the association between each job and its corresponding process, as well as the hierarchical relationship between the master and worker nodes employed in the MapReduce framework. Wilkes (n.d.) The dataset has been employed to facilitate the analysis of resource allocation during job initiation and in the event of failure, thereby aiding in the comprehension of the underlying process. The resilient approach involves the utilisation of Artificial Intelligence to comprehend the allocation of resources prior to the actual allocation process. This approach enhances the dependability of the system by introducing methodologies to anticipate the potential failure of a process prior to its occurrence. In order to integrate this concept, it is necessary to develop a model that can effectively forecast task failure.

Previous research has explored various models that have successfully achieved this objective. Nevertheless, these models face the challenge of over-fitting, a phenomenon commonly associated with the scarcity of training data caused by privacy considerations. Therefore, our study has concentrated on two key factors for predicting task failure. Firstly, we have taken into account the limited availability of data, necessitating the utilisation of public datasets that provide dependable behavioural attributes related to task processing. Secondly, we have adopted an approach that effectively addresses the issue of over-fitting. In this particular task, we have chosen to utilise an ensemble methodology. This approach highlights the combined effect of the outputs produced by multiple models to address the inherent inequalities present in the findings. The ensemble approach can be classified into three distinct categories, namely Boosting, Bagging, and Stacking. Prior research has examined the phenomenon of boosting; however, it is imperative to additionally think about the methodology of stacking. Through the utilisation of stacking, a more

comprehensive visual depiction of our model's performance can be achieved, facilitating the necessary adjustments. The stacking method consists of two separate steps. In the first layer, a selection of multiple models is made, with each model generating individual predictions. The purpose of employing the second layer is to combine the prediction, often referred to as a meta-learner.

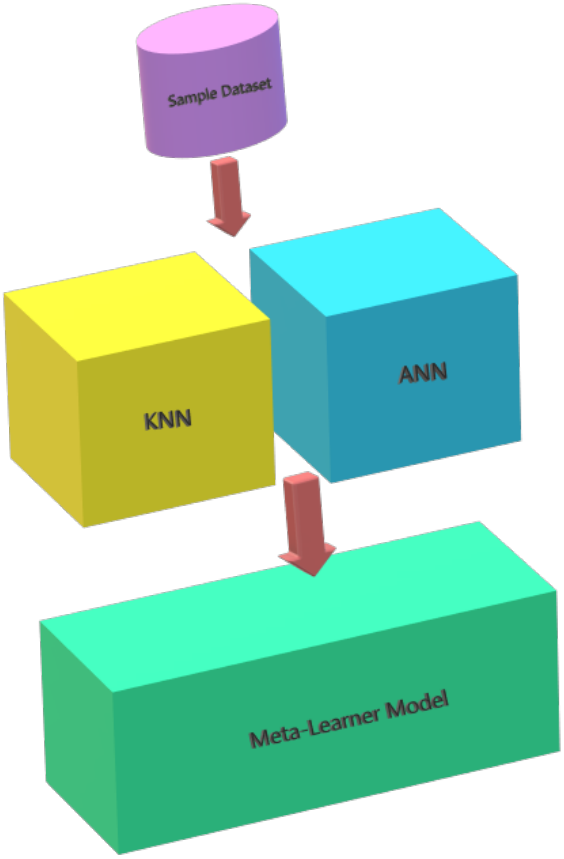

Figure 1: Stacking Method

In the initial layer of our ensemble model, we have included two models: ANN and KNN. The Artificial Neural Network is a versatile computational model that consists of three layers. ANN's are a valuable tool for predicting and forecasting data by emulating human-like behaviour. This process involves the creation of artificial neurons that mimic the structure and function of their biological counterparts. These artificial neurons consist of three essential components: input, transformation, and output. ANN's, or artificial neural networks, possess a notable capability in establishing intricate associations between input and output variables, as well as discerning patterns within the provided dataset. The model exhibits different variations, such as the feed-forward neural network and the Hopfield neural network. These variations differ in terms of the direction in which data

is looped, with the feed-forward network looping data in one direction and the Hopfield network looping data in a symmetric bi-directional manner Vladimir E. (2022).

Although KNN is considered a fundamental machine learning model capable of pattern recognition, it is important to note that this model does not rely on any underlying assumptions about the data. This model facilitates the classification of data by determining its proximity to a given group, thereby assigning it to a distinct cluster IBM (n.d.). For the purpose of this research, both models have been considered as they complement each other in terms of addressing black box issues and over-fitting. KNN is also sensitive to irrelevant features while ANN has a strong inter dependability of feature as it create relationship between features to classify the classes Indyk and Motwani (1998). As previously stated, ANN are considered to be effective models for pattern recognition and classification tasks due to their capacity to establish relationships between input parameters. On the other hand, KNN is a classification algorithm that does not require any specific parameters, but it may exhibit a biased nature in its classification outcomes. Consequently, by combining these two models, it is possible to mitigate the issues of over fitting and bias. This can be achieved by training a second layer of the Ensemble model using the predictions generated by the KNN and ANN . We used Logistic regression in the second layer to combine the result The subsequent phase entails the analysis of the data, as the dataset available is of substantial magnitude, requiring the utilisation of a representative subset. The consolidation of data fields has been undertaken, encompassing a restricted amount of information derived from each of the eight clusters. Each cluster consists of four data tables that store information related to Machine events, Collection events, Instance events, and instance usage Table. The dataset provided for analysis consists of approximately four million records. This dataset incorporates a composite field that encompasses various information pertaining to the requested CPU memory for resource allocation, CPU usage data, shared resource reservation for allocation sets, and the relationship between jobs and their parent entities Wilkes (n.d.)

Lets explain the entire steps for generating a predictive model that uses ensemble approach.

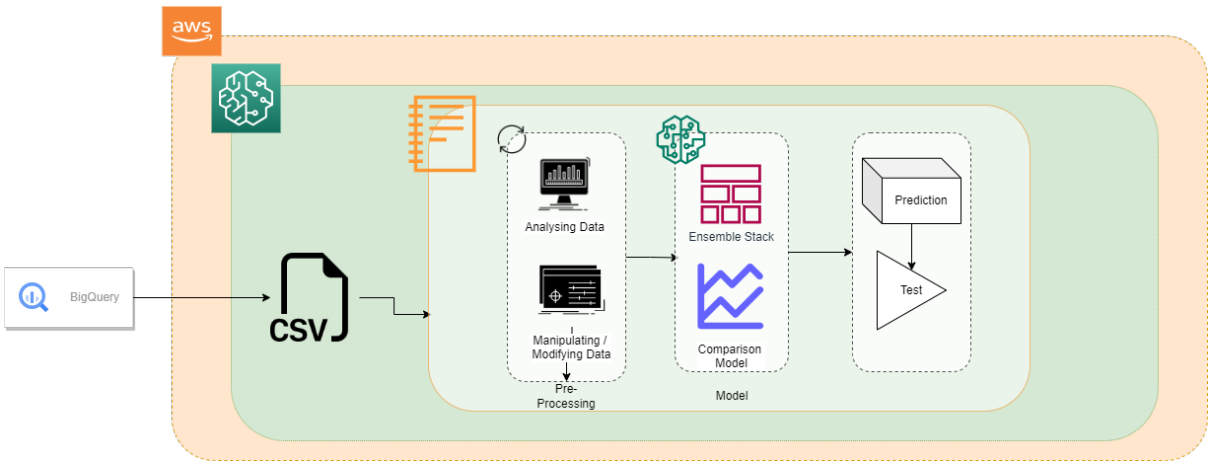

Figure 2: Block diagram of the Methodology

The advent of cloud computing has facilitated the utilisation of cloud resources for the purpose of processing large models, resulting in enhanced computational speed and the optimisation of compute-intensive systems. The ensemble model is a sophisticated

model that necessitates two layers, making the utilisation of cloud computing the most optimal approach for achieving faster and more accurate results. Initially, it is imperative to establish the dataset and subsequently engage in pre-processing procedures. The data exhibits a temporal dimension and demonstrates a significant correlation, wherein the negative correlation is positively associated with task failure and serves as a causal factor for task failure.

During the data pre-processing stage, certain fields that are deemed unnecessary are imputed, and categorical and ordinal values are converted into binary representations (zeros and ones). This conversion facilitates the machine's ability to comprehend and identify patterns within the data. During the processing phase, we establish a design that enables the selection of suitable features from the pre-processed data following a thorough analysis. Next, we proceed to select models that have the potential to enhance the accuracy of our predictions. Considering the factors of over-fitting and the black box phenomenon, we have selected three models. The first layer will be trained using the two models while the second layer will be used to combine the two model's result by training another model to understand the pattern. The two models KNN and ANN are stacked over the Logistic regression model as show in the 1, and using this approach we should achieve a higher prediction accuracy. In the future this model can be packaged and applied to any API which can be further connected to a queuing service that can act as a microservice and provide probability of task failure back to the task scheduling algorithm, to predict the allocation of resources. This approach will benefit data centers and cloud service provider by saving on resource allocation

4 Design Specification

The model is executed utilising the AWS cloud service, which provides us with the ability to execute intricate models without the need to consider resource procurement. SageMaker provides access to compute optimised instances, but our college's limited access prevents us from utilising accelerated compute instances with high GPU capabilities for training deep learning models. Furthermore, we have enhanced the efficiency of our experiment by selecting a subset of the data that can be collected within a limited time period. Due to the utilisation of a deep learning model within a complex methodology, a greater amount of memory was necessitated for the research. Consequently, an instance with a higher memory capacity was selected. The instance is equipped with 96 virtual central processing units (vCPUs), a memory capacity of 192 gigabytes (GB), and does not include any graphical processing units (GPUs). In addition, we are utilising the PyTorch Kernel, which falls within the realm of deep learning containers. The kernel encompasses a comprehensive collection of essential frameworks and libraries, which are conveniently stored within a docker image.

5 Implementation

As outlined in the methodology section, our research will primarily concentrate on the implementation of the model in accordance with our specified research criteria. We have used Python to create the Notebook experiment on AWS SageMaker a services that gives us capability to use instance with high compute power to process the model. The implementation will be divided into four stages, namely Pre-processing of data, Feature

engineering, Prediction, and Comparison of the models. Each stage possesses a distinct purpose, which will be elaborated upon in the subsequent discussion.

5.1 Stage 1 - Pre-Processing

The data extracted from the Google Cluster Trace Version 3 2019 has been transformed into a CSV sample. The data is subjected to analysis in order to detect both null values and numeric values, thereby enabling the identification of fields that contain null values. Next, we proceed to identify all fields classified as numerical, as well as distinguishing those fields that exhibit categorical and ordinal characteristics. subsequently we proceed to assess whether our training attribute contains a balanced distribution of data. It has been observed that the dataset exhibits an imbalance, prompting the need to employ data sampling techniques. The process of sampling the data allows for the training of the model using an equal number of true positives and an equal number of false positives. Utilising this approach will yield enhanced precision in forecasting and comprehending performance metrics, including F1 score, recall, and precision. The subsequent phase of our analysis entails determining the appropriate features to include, a process facilitated by visualising the data through graphical representations such as box plots or histograms.

5.2 Stage 2 - Feature Engineering

This stage primarily emphasises the modification of features and the selection of those that are most appropriate for effectively projecting data. Upon conducting an analysis of the data, our attention is directed towards the development of features and the transformation of the data in order to enhance the performance of the machine learning algorithm. Upon observing the data, it becomes apparent that the requested resource is stored in a JSON format. Consequently, it is necessary to convert the data into multiple columns in order to establish a relationship for the model. This functionality is subsequently employed to eliminate any correlation in order to determine whether the field is suitable for enhancing the model's predictive performance. During the data observation process, it is necessary to scale the numerical fields in order to align their magnitudes with those of other fields. It is necessary to exclude the 'failed' result field in order to effectively train the model. The Standard scalar is employed to normalise the values. In order for the machine to comprehend the cardinal and ordinal values as inputs, it is necessary to convert them into binary representations this is achievd using label encoder function. During this transformation process, it becomes evident which values are most compatible with the model. Once the data has been transformed, we can proceed with predicting the value. This process involves partitioning the dataset into two subsets, with 80 percent allocated for training purposes and the remaining 20 percent reserved for testing purposes.

5.3 Stage 3 - Prediction

The implementation of the model prediction in this stage utilises the Sklearn library. The initialization of the model involves invoking the constructors of the respective models. In order to conduct a comparative analysis, it is necessary to initialise the K-Nearest Neighbours (KNN), Artificial Neural Network (ANN), and logistic regression models as the initial steps in the prediction process. This step involves training the model by fitting it with 80 percent of the available data. All three approaches will be employed in this

activity. The subsequent stage following the fitting of the model involves evaluating the trained model by testing it with a subset comprising 20 percent of the available data. The outcome of this function will be stored in a variable that includes a prefix of "test". The predict function is responsible for conducting the validation of our training model and generating the convolution matrix. This will facilitate the examination of the performance metrics of the models. Next, the Sklearn metrics library is utilised to invoke a static function named "classification report" in order to produce a comparative report between the fitted value and the tested value. The steps involved in implementing the Artificial Neural Network (ANN) model and K-Nearest Neighbours (KNN) model are followed accordingly. However, when working with the Ensemble model, it is necessary to initialise the object with the estimator and final estimator parameter, which is also referred to as the meta learner layer. In the process of initialising an ensemble model, it is necessary to first initialise the Artificial Neural Network (ANN) and K-Nearest Neighbours (KNN) models objects. These initialised models objects should be stored in an array and subsequently passed as an arguments to the estimator parameter of the Stacking classifier object reference. Additionally, the logistic regression initialised object should be passed to the final estimator parameter of the stacking classifier object reference. Once initialized we can follow the above process of fitting and predicting and comparing.

5.4 Stage 4 - Comparison

The final stage is to compare each models performance matrix. This matrix gives us a brief understanding how the model has performed by analysing the F1 score, recall , precision and accuracy, RSME and R square

6 Evaluation

The predictive model took into account a limited number of significant features that were analysed during the data pre-processing stage. The model demonstrates enhanced performance when negative correlation is present, enabling the identification of a limited number of tasks that may fail based on prior requests. To understand the graph, 0 implies successful task while 1 implies failed task. The data provides us with an understanding of the successful and unsuccessful tasks across all eight clusters. The cluster assigns instances to execute a task, and as the number of instances increases, a greater amount of the task is processed. The figure presented below illustrates that clusters 3, 4, and 6 exhibit the greatest number of instances during task execution thus providing a glimpse of number of task executed is more in the these three clusters.

Although our comprehension of memory allocation across all clusters is limited to a basic level. This observation implies that despite the relatively low frequency of occurrences on cluster 8, a significant amount of memory is allocated to the resource. The observed deviation from the norm in behaviour may indicate that the task is necessitating supplementary memory for its processing or that the task is experiencing failures, leading to repeated attempts and subsequently greater memory consumption in comparison to other clusters. The examination of memory allocation is a pivotal factor in our anticipation of task failure. There exists an inverse correlation between the allocation of memory and the occurrence of abnormal behaviour within the cluster. Upon careful examination of the dataset, it becomes evident that a greater amount of memory is

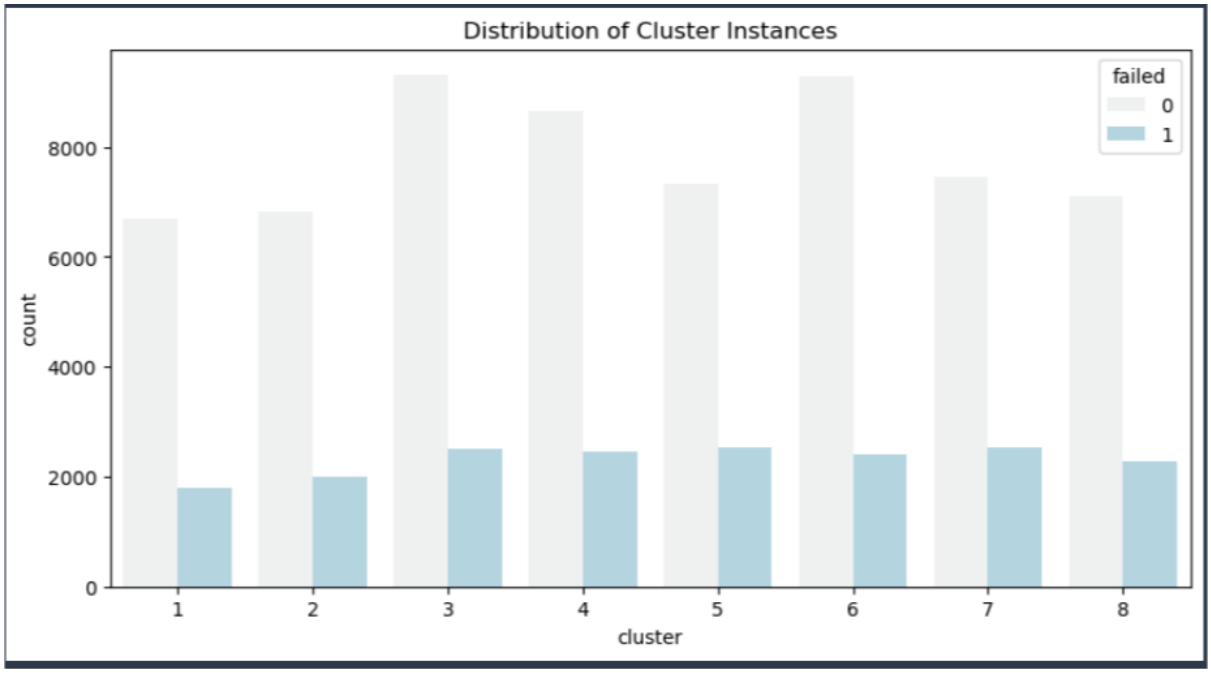

Figure 3: Distribution of Cluster instances

typically allocated to terminated resources. Conversely, a majority of the failed tasks exhibit a higher level of input/output (I/O) consumption, indicating a prevalence of disc write failures. Moreover, it can be inferred from our comprehension that the primary cause of task failures within a cluster is resource failure. This observation underscores the significance of incorporating resource utilisation and allocation as key factors in the development of the predictive model.

This observation has been instrumental in the process of feature selection. We have identified the availability of requested resources that provide information regarding the required memory for a task, as well as the duration of each task. These observations have been recorded at intervals of 5 minutes, allowing us to analyse the time taken for task completion and the subsequent time required for task retries. The aforementioned factor will assume a pivotal role in comprehending the occurrence of task failure and facilitating the anticipation of task failure.

A comparative study has been done between few models to show the best fit model for prediction of Task Failure. This study also will also provide an insight on the best model for this kind of data. Since we are focusing on classifying a task between success and failure we have used two ensemble model results and two machine learning model to provide a comparative study between both the approaches. We have also consider factors such as F1 score, Accuracy , and recall to get a deeper understanding of which model is best suited for such predictions. We have split the data into 80 percent training data and 20 percent testing data to validate the prediction. Then we measure the model's performance Let us define the performance matrix below: **Precision**: It is defined as the accuracy of the classifier to predict positive labels Davis and Goadrich (2006) **Recall**: It is defined as the ability to correctly identify all instance of a given class Powers (2020) **F1-Score**: It is the harmonic mean of precision and recall Goutte and Gaussier (2005) **Accuracy**: It is the holistic metrics , that represents the proportion of all correct prediction over total predictions. Japkowicz and Shah (2011)

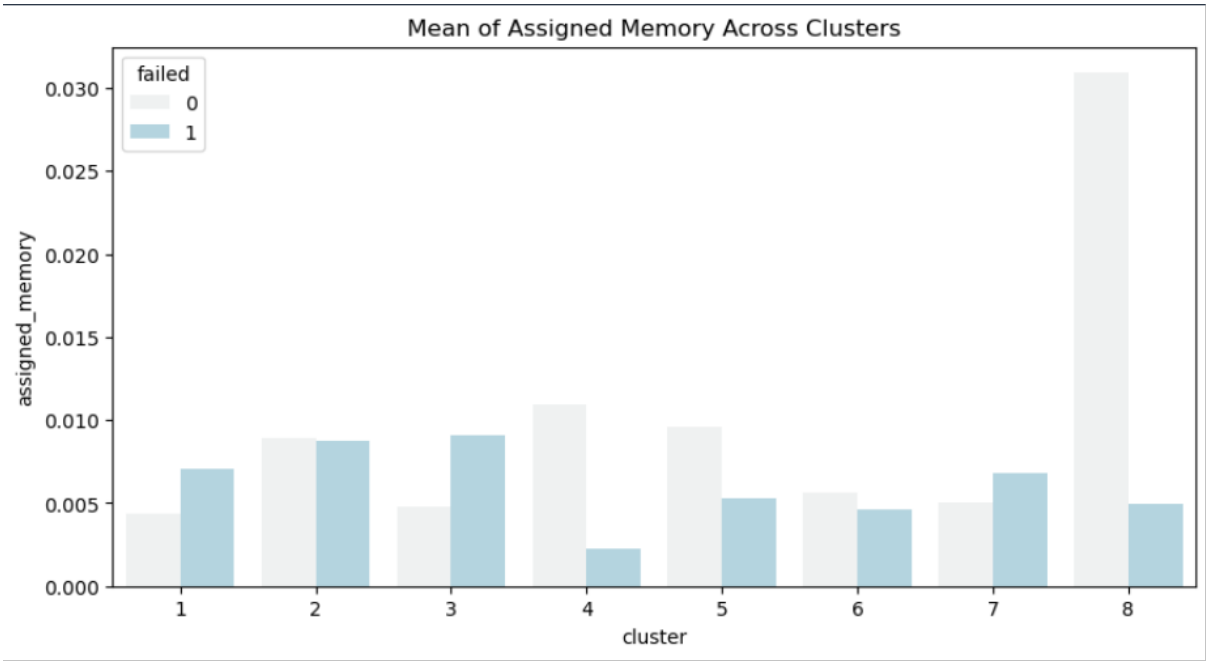

Figure 4: Mean memory across clusters

6.1 Experiment / Case Study 1

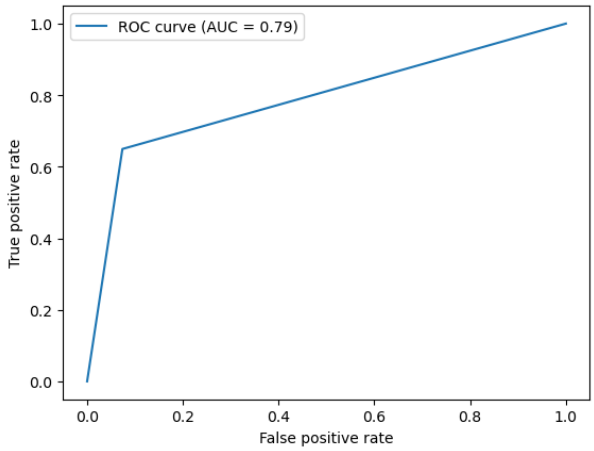

Figure 5: KNN AUC Curve

In this experiment we are going to discus the result of the KNN Model with respect to classify the success or failure of the task . The reported accuracy, which stands at approximately 0.863, signifies that the model effectively classifies around 86.3 percent of the instances. While the commendable accuracy rate of the predictions is evident, there appears to be some inconsistency when considering the subsequent metrics of precision, recall, and F1-score, which exhibit a near-perfect performance. The existence of these disparities calls for a thorough investigation into both the assessment procedure and the associated data. The RMSE value of 0.1044, when evaluated within the framework of the target variable's scale, offers significant insights into the precision of the predictive model.

```
Accuracy: 0.8632052229613205
RMSE: 0.10441120594117363
R_Squared_value: 0.9380546251096766
              precision    recall  f1-score   support

           0       1.00      0.99      0.99     12535
           1       0.96      1.00      0.98      3701

    accuracy                           0.99     16236
   macro avg       0.98      0.99      0.98     16236
weighted avg       0.99      0.99      0.99     16236
```

Figure 6: KNN Model performace Metrics

Typically, lower values of root mean square error (RMSE) are indicative of higher levels of predictive accuracy. The R Square value of 0.938 suggests that the model accounts for approximately 93.8 percent of the variability observed in the dependent variable. This observation implies that the model exhibits a robust capability for offering thorough explanations.

- Class 0
  - .When the value of class 0 reaches 1.00, its precision is regarded as impeccable. The obtained score demonstrates the model's consistent ability to accurately classify instances as class 0.
  - The model exhibits a notable degree of precision in detecting instances belonging to class 0, accurately categorising 99 percent of such instances when employing a threshold of 0.99. This finding demonstrates the efficacy of the approach in reducing the incidence of false negatives within this specific category.
  - The F1-score of 0.99 obtained for class 0 serves as a confirmation of the model's capacity to appropriately balance precision and recall.
- Class 1
  - The model exhibits a notable degree of precision in forecasting class 1, with a minimal margin of error, as evidenced by its precision score of 0.96.
  - The recall score of 1.00 demonstrates the model's proficiency in correctly classifying all instances associated with class 1.
  - The F1-score of 0.98 obtained for class 1 provides further evidence of the model's efficacy in striking a balanced compromise between precision and recall.

6.2 Experiment / Case Study 2

Now let us consider a more complex model . The ANN is a deep learning model that has three layer . This model represent the structure of Neuron in the brain and it has an input layer , a middle layer and an output layer. Similar to a neuron data is passed across each layer. Moreover a nodes importance's is based on the inputs associated weights that can be negative or positive. The neuron is said to be active if the weight is positive and

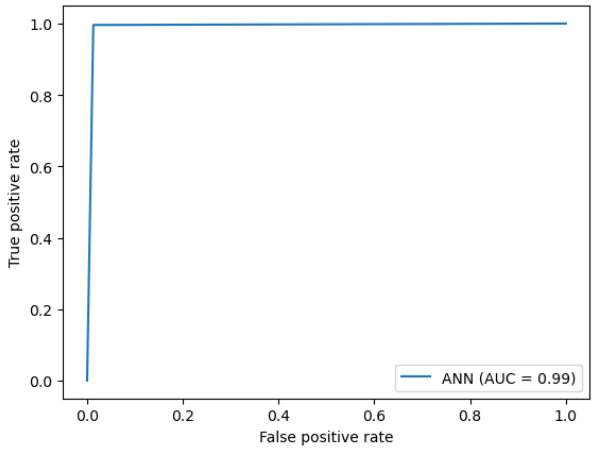

Figure 7: ANN AUC

```
Accuracy: 0.9890983000739099
RMSE: 0.10441120594117363
R_Squared_value: 0.9380546251096766
              precision    recall  f1-score   support

           0       1.00      0.99      0.99     12535
           1       0.96      1.00      0.98      3701

    accuracy                           0.99     16236
   macro avg       0.98      0.99      0.98     16236
weighted avg       0.99      0.99      0.99     16236
```

Figure 8: ANN Performance metrics

inactive if the weight is said to be negative . The result encompass a range of metrics for both regression and classification tasks, providing a comprehensive understanding of the effectiveness of the utilised model. A precision rate of approximately 0.989 indicates that the model accurately categorises approximately 98.9 percent of the instances, which is a significant accomplishment in various fields, particularly when considering the intricacy of real-world data. A root mean square error (RMSE) value of 0.1044 indicates the magnitude of the standard deviation of the errors in the predictions. When placed within the context of the magnitude of the target variable, this provides valuable information regarding the predictive precision of the model. In general, a decrease in the root mean square error (RMSE) is indicative of improved predictive accuracy. A value of R Square close to 0.938 indicates that the model explains approximately 93.8 percent of the variance in the dependent variable. This highlights the considerable explanatory capacity of the model. Now let us consider the generated classes and discuss about the Precision, Recall, F1-Score in the two classes .

- Class 0:
  - The precision metric, denoted by a value of 1.00, indicates that the model's accuracy in correctly predicting class 0 is 100 percent.
  - The recall metric, denoted as 0.99, signifies that the model accurately classifies 99 percent of the instances belonging to class 0.
  - The F1-Score of 0.99 indicates a favourable equilibrium between precision and recall.
- Class 1:
  - The precision score of 0.96 indicates that 96 percent of the instances classified as class 1 are correct, with a small chance of misclassification.
  - The perfect recall score of 1.00 for class 1 demonstrates the model's proficiency in correctly identifying and capturing all instances belonging to this particular class.
  - The F1-Score of 0.98 demonstrates the model's ability to effectively balance precision and recall for class 1.

6.3 Experiment / Case Study 3

The ensemble model employed in this study utilises a stacking approach to address the limitations of both KNN and ANN, as previously discussed. The complexity of this model renders it advantageous in comparison to alternative models. However, the model's complexity necessitates a substantial investment of time and computational resources for both training and testing. Another aspect that warrants consideration in the observation is the utilisation of intricate models, such as ANN, which consist of multiple layers. Utilising this approach is not recommended due to the significant increase in complexity of the overall model, resulting in extended waiting times. The reported accuracy of approximately 0.9913 suggests that the model attains a classification accuracy rate of approximately 99.13 percent. The model demonstrates a strong and efficient performance, as evidenced by its high level of precision. The RMSE value of 0.09319, when considered in relation to the magnitude of the target variable, indicates that the model exhibits a

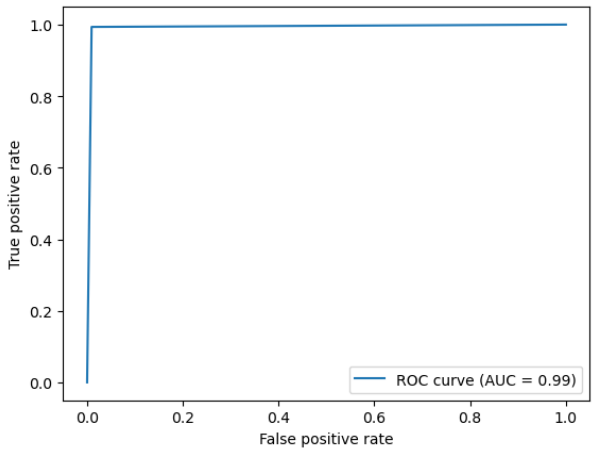

Figure 9: Ensemble Model AUC

```
Accuracy: 0.9913155949741316
RMSE: 0.09319015519822059
R_Squared_value: 0.9506536844094035
              precision    recall  f1-score   support

           0       1.00      0.99      0.99     12535
           1       0.97      0.99      0.98      3701

    accuracy                           0.99     16236
   macro avg       0.98      0.99      0.99     16236
weighted avg       0.99      0.99      0.99     16236

 1435.63 seconds
```

Figure 10: Ensemble Metric

[779]:

	0	1	max_prob
0	0.999019	0.000981	0.999019
1	0.999042	0.000958	0.999042
2	0.999037	0.000963	0.999037
3	0.005524	0.994476	0.994476
4	0.997742	0.002258	0.997742
...	...	...	...
16231	0.998167	0.001833	0.998167
16232	0.998727	0.001273	0.998727
16233	0.998844	0.001156	0.998844
16234	0.998928	0.001072	0.998928
16235	0.991453	0.008547	0.991453

16236 rows × 3 columns

Figure 11: Ensemble probability

notable level of accuracy in predicting outcomes in regression analyses. Typically, models with lower root mean square error (RMSE) values are considered to exhibit superior performance. A value of 0.9507 for the coefficient of determination (R Square) indicates that approximately 95.07 percent of the variability in the dependent variable can be explained by the independent variables. The model's significant explanatory ability is emphasised by the high coefficient. Now let us consider the generated classes and discuss about the Precision, Recall, F1-Score in the two classes

- Class 0
  - The precision score of 1.00 achieved for class 0 indicates that the model's predictions for this particular class exhibit a high level of accuracy.
  - The recall value of 0.99 demonstrates the model's proficiency in correctly identifying 99 percent of all instances belonging to class 0, thereby highlighting its effectiveness in reducing the occurrence of false negatives.
  - The F1-score of 0.99 achieved for class 0 indicates a well-balanced trade-off between precision and recall, indicating the model's consistent and reliable performance.
- Class 1:
  - The precision value of 0.97 assigned to class 1 demonstrates a notable level of accuracy, albeit with a small potential for error.
  - The recall value of 0.99 indicates that the model has a high level of proficiency in accurately identifying almost all instances belonging to class 1.
  - The F1-score of 0.98 achieved for class 1 highlights the model's ability to effectively balance precision and recall.

6.4 Discussion

As previously stated, an examination of the performance of each model allows for a comprehensive assessment of the various factors at play in each model. In order to utilise any of the models, it is imperative to possess an in-depth understanding of the data. The performance of a model is determined by the data, and the process of feature selection significantly influences the model's predictive capacity. Using the performance metrics we were able to provide further insights on the performance of each model. We could notice quite high Accuracy , F1-Score and recall. This signifies that the model is really optimal in classifying the task success or failure however let us discus the drawback of the result.

- Possible Conflicts
  - The ANN model is prone to over-fitting due to its excessively high accuracy, indicating that the model has the capacity to memorise patterns rather than truly learn them. The KNN algorithm is also susceptible to the same issue. However, an ensemble model incorporates a meta learner layer to aggregate the results, thereby significantly reducing the likelihood of over-fitting.

- Furthermore, we have conducted training on a model that is specifically tailored to Google Cloud Trace and optimised for enhancing Google's cluster performance. However, it is important to consider a more comprehensive approach where the model can be trained to handle various types of tasks that are submitted as jobs to the cloud.
- During the experimental phase, it is imperative to incorporate unseen data in order to evaluate the model's ability to accurately classify the given task.
- The aforementioned outcome reveals that all three models exhibited a slight decline in performance when classifying Class 1, thereby raising concerns regarding the distribution of classes in the dataset and indicating an imbalance of data.
- Finally, the Ensemble model exhibits dependable outcomes; however, it possesses a higher level of complexity due to the amalgamation of two machine learning algorithms. Additionally, we have incorporated a Neural network, which further contributes to the intricate nature of the ensemble model, resulting in increased resource requirements compared to alternative models.

Other than these issue the ensemble model is subject to privacy issue due to the lack of transparency

Now let us discuss about the current three models

- KNN Model
  - Accuracy: 0.8632052229613205
  - RMSE: 0.10441120594117363
  - R-Squared: 0.9380546251096766
  - Classification Metrics: The model gave a nearly perfect precision, recall and F1 score for both classes however there were visible anomaly between the accuracy and other metrics prompting towards issues during pre processing of the data
- ANN Model
  - Accuracy: 0.9890983000739099
  - RMSE: 0.10441120594117363
  - R-Squared: 0.9380546251096766
  - Classification Metrics: The model shows high precision, recall, and F1-score for both classes, with little to no difference in the accuracy. This also can signify that the model is over fitting and can be only validated by passing unseen data to verify the model.
- Ensemble Model
  - Accuracy: 0.9913155949741316
  - RMSE: 0.09319015519822059
  - R-Squared: 0.9506536844094035
  - Classification Metrics: The model shows higher precision, recall, and F1-score for both classes, compared to the ANN and KNN individually. The lower RMSE shows that the model has better accuracy compared to the ANN.

7 Conclusion and Future Work

Based on the aforementioned experiments, it can be deduced that employing an ensemble model yields a dependable predictive model with 1 to 15 percent improvement in accuracy compared to ANN and KNN respectively. Nevertheless, it is important to acknowledge that the model does possess certain limitations. The Ensemble model is expected to exhibit superior performance when utilising two machine learning models jointly, as opposed to single deep learning model. Alternatively, one could employ a deep learning model, such as an ANN or a Deep Learning algorithm, by reducing the number of features and normalising the data. This approach aims to enhance the model's performance. The training of two models in the ensemble model often necessitates a significant amount of memory, indicating that the ensemble model relies on compute-intensive resources. When delving further into the subject, it becomes evident that increasing the complexity of the model leads to a longer duration for training and testing. Additionally, leveraging cloud services like SageMaker can facilitate the construction of the model by providing enhanced computational capabilities and improved efficiency. This model provides us with enhanced understanding for selecting an ensemble model, contingent upon the consistent utilisation of batch jobs as the primary means of task execution within our workforce. The proposed implementation would facilitate the development of task scheduling algorithms within data centres. These algorithms would utilise an application programming interface (API) to forecast the likelihood of task failure. Based on this prediction, the task scheduler would allocate resources according to a priority determined by the scheduler storing the task in queue, provided that the predicted value falls within a specified range. The incorporation of such a framework would be highly advantageous, as its integration into a task scheduling algorithm would provide researchers with the opportunity to enhance the algorithm through the application of machine learning techniques. This would facilitate further investigation into green computing, specifically focusing on the utilisation of a resilient approach for fault tolerance.

References

Agarwal, K. K. and Kotakula, H. (2022). Fault tolerance in cloud: A brief survey, *Springer eBooks* pp. 578–589.

Alahmad, Y., Daradkeh, T. and Agarwal, A. (2021). Proactive failure-aware task scheduling framework for cloud computing, *IEEE Access* **9**: 106152–106168.

Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and roc curves, *Proceedings of the 23rd international conference on Machine learning*, pp. 233–240.

El-Sayed, N., Zhu, H. and Schroeder, B. (2017). Learning from failure across multiple clusters: A trace-driven approach to understanding, predicting, and mitigating job terminations, *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)* .

Gao, J., Wang, H. and Shen, H. (2022). Task failure prediction in cloud data centers using deep learning, *IEEE Transactions on Services Computing* **15**: 1411–1422.

Goutte, C. and Gaussier, E. (2005). A probabilistic interpretation of precision, recall and f-score, with implication for evaluation, *European conference on information retrieval*, Springer, pp. 345–359.

IBM (n.d.). What is the k-nearest neighbors algorithm? — ibm.
URL: *https://www.ibm.com/topics/knn*

Indyk, P. and Motwani, R. (1998). Approximate nearest neighbors: towards removing the curse of dimensionality, *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pp. 604–613.

Japkowicz, N. and Shah, M. (2011). *Evaluating learning algorithms: a classification perspective*, Cambridge University Press.

Jassas, M. S. and Mahmoud, Q. H. (2021). A failure prediction model for large scale cloud applications using deep learning, *2021 IEEE International Systems Conference (SysCon)* .

Marahatta, A., Xin, Q., Chi, C., Zhang, F. and Liu, Z. (2021). Pefs: Ai-driven prediction based energy-aware fault-tolerant scheduling scheme for cloud data center, *IEEE Transactions on Sustainable Computing* **6**: 655–666.

Powers, D. M. (2020). Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation, *arXiv preprint arXiv:2010.16061* .

Ragmani, A., Elomri, A., Abghour, N., Moussaid, K., Rida, M. and Badidi, E. (2020). Sciencedirect the 10th international symposium on frontiers in ambient and mobile systems (fams 2020) adaptive fault-tolerant model for improving cloud computing performance using artificial neural network sciencedirect the 10th international symposium on frontiers in ambient and mobile systems (fams 2020) adaptive fault-tolerant model for improving cloud computing performance using artificial neural network, *Procedia Computer Science* **170**: 0–000.

Shahid, M. A., Islam, N., Alam, M. M., Mazliham, M. and Musa, S. (2021). Towards resilient method: An exhaustive survey of fault tolerance methods in the cloud computing environment, *Computer Science Review* **40**: 100398.

Tengku Asmawi, T. N., Ismail, A. and Shen, J. (2022). Cloud failure prediction based on traditional machine learning and deep learning, *Journal of Cloud Computing* **11**.

Vani, K. and Sujatha, S. (2022). A machine learning framework for job failure prediction in cloud using hyper parameter tuned mlp, *2022 Second International Conference on Advanced Technologies in Intelligent Control, Environment, Computing amp; Communication Engineering (ICATIECE)* .

Vladimir E., B. (2022). Artificial neural networks.
URL: *https://research.ebsco.com/linkprocessor/plink?id=92b3c34c-2d5b-3a26-920e-20625ccacedc*

Wilkes, J. (n.d.). Google cluster-usage traces v3.