InOt-RePCoN: Forecasting User Behavioural trend in Large-Scale Cloud Environments

John Panneerselvam, Lu Liu and Nick Antonopoulos

Department of Engineering and Technology, University of Derby,
Derby, United Kingdom, DE22 1GB
Email: {j.panneerselvam; l.liu and n.antonopoulos}@derby.ac.uk
Corresponding Author: Lu Liu

Abstract

Cloud Computing has emerged as a low cost anywhere anytime computing paradigm. Given the energy consumption characteristics of the Cloud resources, service providers are under immense pressure to reduce the energy implications of the datacentres. Forecasting the anticipated future workloads would help the service providers to achieve an optimum energy-efficient scaling of the datacentre resources in accordance with the incoming workloads. But the extreme dynamicity of both the users and their workloads impose several challenges in accurately predicting their future behavioural trend. This paper proposes a novel prediction model named InOt-RePCoN (Influential Outlier Restrained Prediction with Confidence Optimisation), aimed at a tri-fold forecast for predicting the expected number of job submissions, session duration for users, and also the job submission interval for the incoming workloads. Our proposed framework exploits autoregressive integrated moving average (ARIMA) technique integrated with a confidence optimiser for prediction and achieves reliable level of accuracy in predicting the user behaviours by the way of exploiting the inherent periodicity and predictability of every individual jobs of every single users. Performance evaluations conducted on a real-world Cloud trace logs reveal that the proposed prediction model outperforms the existing prediction models based on simple auto-regression, simple ARIMA and co-clustering time-series techniques in terms of the achieved prediction accuracy.

Keywords: Energy efficiency, Cloud datacentres, User behaviours, Confidence optimisation, Resource Scaling

1. Introduction

Cloud Computing has emerged as a prominent service paradigm of low-cost anytime and anywhere computing for various business needs. Despite the tremendous outreach of Cloud Computing in various application domains, Cloud datacentres are also witnessed to be one of the major consumers of energy [1-3] and as environmental pollutants. This energy consuming characteristics of Cloud datacentre resources necessitate the demand for promoting green computing [4] ultimately to reduce the energy related implications of Cloud datacentres. One of the possible ways of achieving energy efficiency in Cloud datacentres is to predict [5] the future workload demands, thereby maintaining the resource utilisation [6-8] under the desired level of energy consumptions. But this involves various barriers, since wrong prediction would significantly affect Cloud datacentre management. There is a higher possibility of SLA (Service Level Agreement) violations [9] with wrong prediction results, which directly affects the Quality of Service (QoS) of the providers by not satisfying the Quality of Expectations (QoE) of the users. Workloads arriving at the datacentres are scheduled on to the Virtual Machines (VMs) deployed on the physical servers for processing based on the job requirements. Users often exhibit varied service requirements for their job submissions such as reduced job turn-around time, scheduling priority, resource constraints etc. Modelling the behaviours of the Cloud workloads closely correlated with the user behaviours is a challenging task, since a computational model cannot identically reflect the human behaviours.

An efficient prediction model in Cloud Computing should incorporate the relationship existing among the job submissions and the user behaviours. Some of the important prediction metrics inherent among the job and user behaviours include job submission time, submission frequency, user session duration, user requested resource levels etc. Most of these metric parameters exhibit both temporal and/or spatial variations and correlations, which could be both significant positives (maximum correlations) and significant negatives (minimum correlations). Significant positives represents the persistence of a system metric to remain consistent over a period of time. The degree of such positive and negative correlations should be carefully incorporated in prediction modelling, since clusters of significant positives lead to effective prediction analysis whereas clusters of significant negatives affects the prediction accuracy to an irresistible margin. These correlation metrics exhibits dynamic shifts in time, as the workloads usually fluctuate in time driven by the user behaviours. Identifying the positive correlations among the Cloud workloads and user behaviours over time helps extracting the hidden periodicity among the Cloud entities. Periodicity pattern [10] identifies the recurring behaviours among both the Cloud users and their job submissions. Such a Cloud periodicity can be defined in relation with various time-bound periodical effects

[11] such as time-of-the-day, day-of-the-week, week-of-the-month and month-of-the-year effects etc. Time-of-the-day effect defines the correlation of the user behaviours with different business hours of a day. Such correlations are usually evident across the user-driven job events at the Cloud datacentres. For instance, Cloud datacentres might face an increased number of job submissions during the peak business hours and a declining number of job submissions during off-peak business hours. Similarly, day-of-the-week effect is the day-wise correlated user behaviours where the job event correlations are evident across the representative days of different weeks. Usually Cloud providers face an increasing number of users and job submissions during weekdays [12] and both decline during weekends.

An integral requirement of an effective prediction model is the characterisation of the workload and the user behaviours. The dynamic nature of the Cloud users and their workloads demand an extensive and continuous analysis for characterising the user behaviours in relation to the operating business hours. Users of Cloud services generally co-exists from different business context and submit workloads of diverse resource requirements. A single user might submit jobs of different types under dynamic arrival frequency during a given session. This user session is the duration occupied by the users at a Cloud datacentre during a period of observation. Users are the actual drivers of the Cloud workloads, thus validating the relationship [13] between user behavioural trend and their corresponding job submissions is crucial in prediction modelling. Existing works of prediction model [2, 11, 14-20] aimed at forecasting job arrival trend in Cloud environments include SPAR - a periodic autoregressive algorithm, RPPS - a simple ARIMA forecast, multiple time series approaches, linear regression, neural networks, Markov based approaches, clustering approaches, Bayesian models etc. Despite the existing works of prediction models in Cloud Computing to date, there is still a lack of an effective prediction model that can capture the inherent characteristic diversity and the correlations between users and their jobs submission trends. Further, most of such approaches are focused only on characterising the workload behaviours leaving the user behaviours unnoticed. To this end this paper proposes a novel prediction model named InOt-RePCoN (Influential Outlier Restrained Prediction with Confidence Optimisation) aimed at a tri-fold forecast of the User behaviours, forecasting the anticipated number of job submissions in a session, session duration anticipated for users along with predicting their job submission trend in terms of the submission interval of consecutive submissions of the same jobs from the users. This tri-fold forecast of the user behaviours helps the service providers with a pro-active datacentre management for the purpose of achieving an optimum energy-efficient scaling of the server resources in accordance with the arriving workloads. Our proposed prediction model exploits both the time-of-the-day and day-of-the-week periodicity effects for characterising the user periodicity and predicts the future user behaviours based on a confidence optimised ARIMA forecast. Our proposed model uniquely analyses every single jobs belonging to the users to achieve a reliable level of prediction accuracy. The important contributions of this paper include the following.

1. Analysis and extraction of the predictive features of both users and their corresponding job submissions to build the predictability profiles of users and jobs. By the way of exploiting the periodicity effects, our proposed model computes the predictability weights for every single jobs submitted by the users. This predictability weight has been exploited by our proposed model to reduce the average prediction error by uniquely treating jobs and users characterising different predictability weights.
2. A tri-fold prediction of user behaviours in terms of their job submission trends in Cloud environments. Firstly, forecasting the number of expected submissions of jobs for the target users. Secondly, forecasting the session duration for the anticipated users and finally, predicting the job submission interval of consecutive submissions of the same job from the users for an observed session.

The remainder of this paper is organised as follows: Section 2 reviews the existing prediction models in Cloud Computing to date. Section 3 is covered with a background study on energy efficiency in Cloud Computing, along with revealing the predictability characteristics of both Cloud users and workloads and the dynamic nature of Cloud Computing. Our proposed prediction framework is described in Section 4, with Section 5 describing the prediction mechanism. Section 6 validates our proposed prediction model and Section 7 presents our performance evaluations. Section 8 concludes this paper along with our future research directions.

2. Related Works

Predictive analytics are being carried out in Cloud environments for various purposes such as resource scaling, workload allocation, optimising elasticity etc. In general, there are two important phases of prediction analytics in Cloud environments for energy efficiency, firstly forecasting the anticipated intensity of the arriving job submissions and secondly estimating the resource consumption levels of the arrived jobs. While the former

benefits efficient scaling of the datacentre resources, the later helps with optimum level of resource allocation for the incoming jobs. An approach for clustering tasks [2] of similar characteristics has been proposed to estimate the resource requirements of newly arriving tasks by analysing clusters formed of historical information. This analysis further presents that only 20% of tasks are exhibiting periodicity, the degree of periodicity is crucial in determining the overall prediction accuracy. Since the incoming job trend exhibits better periodicity than tasks and tasks are actually contained within jobs, a hybridised clustering approach of both jobs and tasks might deliver a better estimation of resource requirements for workload execution. *k-means* clustering [21] is a well-known classification approach used for clustering observations, which divides *n* observations into *k* clusters based on the chosen parameters. The clusters are usually formed around the optimal centroids, but determining the optimal number of clusters is often complex. Another complexity prevailing in adopting *k-means* algorithm for classifying Cloud variables is that the algorithm does not scale well for global clusters and clusters of different size and density. Both these complexities necessitates analysing the characteristics of the incoming workloads before choosing the optimum number of clusters. With the Cloud workloads being increasingly heterogeneous, uniquely analysing the incoming workloads to choose the number of clusters might be tedious, and cluster selection based on qualitative metrics of the workloads generally introduce subjectivity in the computation accuracy.

A linear regression based prediction model (LRM) [16] has been proposed for benefitting autonomous resource scaling in Cloud datacentres, by predicting the number of service requests expected at the next interval based on an observation of linear trend of workloads in a relatively short period of time. Though LRM exhibits a lower prediction error, a simple linear approach may not scale well under fluctuating and dynamic workload arriving pattern in a longer time frame. Forecasting the workloads in a relatively shorter term may not allow enough time-scale for resource management due to the wake-up latencies of the machines. Generally in a simple LRM, the mean of the independent variable is expected as a linear combination of the regression coefficients and the predictor variables and this mean is strictly linear due to fixed predictor values. This might prevent the regression from accurately modelling the dependence between the dependent and independent variables under dynamic workload fluctuations.

A Pattern matching workload prediction framework [22] for forecasting the resource usage patterns has been proposed by exploiting historical usage patterns those similar to the current trend. This framework identifies similar usage patterns from the past using Knuth-Morris-Pratt (KMP) algorithm for string matching, but jobs with similar characteristics not necessarily exhibit similar resource usage patterns. Furthermore, task failures within a single job execution significantly affect the overall resource usage levels of the workloads and tasks failure rates vary dynamically during different execution instances. In addition, this algorithm calculates the acceptable error based on the desired number of matches, more matches are usually identified for larger acceptable error. Since the error margin is unique for every prediction match, deciding the trade-off between the desired number of matches and prediction accuracy is always questionable. Enhancing the precision of prediction usually restrain the number of matches identified by the algorithm, thus may not provide suffice historical evidences for estimating the future usage patterns. Addressing the error margin issues of this KMP algorithm, an improved KMP algorithm in combination with a linear regression model [23] has been proposed for load prediction. This model apparently chooses the linear regression model when the workload fluctuation is low, and chooses the KMP model when the incoming trend of workloads exhibits higher fluctuation. This scheme of alternative prediction model may fit the dynamicity of Cloud Computing. Though, it is not guaranteed that the observed current trend stays unaltered during the prediction time, thus the switching scheme may not scale well when the observed trend shifts suddenly in shorter time. A pattern matching scheme [24] for CPU workload sequence forecast has been proposed based on the KMP algorithm. This scheme benefits from a pre-processing phase of data analysis encompassing a time series analysis of the monitored data followed by a Kalman filter to approximate the true data based on observed data. In general, KMP algorithm suffers limitations such that the traditional approach can only match absolute values. Since the incoming traffic is actually sequence of data points, traditional KMP algorithm may not scale well for time series sequential data analysis.

An exponential smoothing (ES) [25] based prediction mechanism has been proposed to predict the future job arrival trend using historical information. ES approach, which predicts the trend of a time series, has been applied in this model to predict the attributes of the future trend by concurrent iterations. Though this approach benefits from the historical trend, limited usage of the historical traces in time may not provide suffice inferences, which necessitates the need of storing historical traces longer incurring additional storage costs. Since Cloud workload behaviours are bound to business hours, contradictions and inaccuracies might arise whilst using the peak time current iterations to predict the off-peak time future trend and vice versa.

A Hidden Markov Model (HMM) based prediction [11] of workload patterns has been proposed by exploring the temporal correlation among the workload behavioural changes, treating workload samples as time series. This model exploits the cross VM correlations resulting from the dependencies among the applications running in different VMs. This prediction framework is aimed at forecasting the workloads on individual VMs based on the workloads groups witnessed in the previous process cycle. Analysing the workloads at the group level to predict the workload pattern on the individual VMs may not deliver precise prediction results. Also predicting the workload patterns should incorporate the knowledge of the user behavioural patterns, since users are the actual drivers of the workloads. But, workload patterns on individual VMs are usually driven by the scheduling and job allocation mechanisms of the service providers not users. An off-line prediction [17] has been conducted based on a pre-recorded resource usage data to forecast short-term resource usages based on a Markov chain model. Markov matrix [26] has been used to predict the future state distribution vector from the current state. In general Markov based approaches work with the fact that estimation of workload patterns exhibit prediction probabilities and the highest probability will lead to an effective prediction result. In general deterministic approaches might deliver better prediction accuracy than probabilistic approaches under the dynamic and timely varying nature of Cloud Computing.

The mean load prediction over a long-term interval has been proposed based on [27, 28] Bayesian model. With an estimated mean load at a given time, this model predicts the mean load into the future time for up to 16 hours. The mean load over consecutive time intervals is estimated an exponentially segmented pattern for the purpose of characterising the host load over a definite period of time. These prediction segments are transformed into a pattern with the heuristics that host load appears with higher correlation among the adjacent short term intervals. This may not necessarily be true in most of the occasions, since Cloud workloads and active users exhibit significant variations between peak and off-peak business hours. Thus long-term prediction based on adjacent segments might not be effective to deliver reliable level of prediction accuracy. Bayesian model works by the way of relating the prior and posterior event probabilities for computing the conditional probabilities. Furthermore, our previous works on evaluating the efficiencies of HMM and Naïve Bayes model in predicting Cloud workloads revealed that both the models are susceptible [5] to an increased error percentage whilst predicting CPU and memory intensive Cloud workloads.

A predictive model based on a degree two polynomial regression [29] has been proposed for benefitting resource scaling in the datacentres. This model estimates the static and dynamic resource requests at the web server tier and the database tier for predicting the optimal configuration required for dynamically varying workloads in order to achieve an optimum provisioning of resources. This prediction model is built using the application performance statistics obtained while the application is still running. In general, polynomial regression models the relationship between the independent and dependent variables based on their non-linear dependence, since the polynomial functions are non-local the value of the independent variable strongly depends on the dependent variable. This prediction model works on real-time based on continuous iterations for estimating the over-provisioning of resources, this necessitates consistent monitoring of the workload intensity and resource consumption profiles which may impose additional overheads in the overall resource provisioning system. Furthermore, this model may suffer from complex time-cost since the overall prediction time is an accumulation of the status response time, process iteration and resource estimation time.

A modified best fit policy [30] has been proposed by treating physical machines as bins and virtual machines as items to be allocated onto the bins for the purpose of minimising the number of active physical servers. Before allocating VMs on to the physical servers, the future load anticipated on the physical machines is computed based on the load on the VMs to be allocated. This algorithm is merely a computation based on the current load rather than a prediction, since the estimation is based on the known VM load. Estimating the anticipated incoming traffic a priori might help better resource management, rather than computing the anticipated load on the servers since this necessitates an additional computation time before the workloads can be allocated for processing. A virtual resource scheduling prediction scheme [31] has been proposed based on Support Vector Machine (SVM). The virtual resource requirements have been estimated by modelling the non-linear relationship between the inputs of the SVM. This model benefits by reconstructing the phase of the system as a time sequence. Phase is a state of the system at a given time, reconstructing the state as a time sequence might help modelling the linear dependence among the consecutive data values which can lead better prediction. But the degree of dependence among the consecutive values will dominate the prediction, and the efficiency of the SVM usually depends on the associated algorithm in learning the relationship inherent among the data values.

SPAR [14] (Spare Periodic Auto-Regression), an autoregressive based prediction model, has been proposed to forecast the workload patterns with the assumption that the dependent variable is highly correlated at every step under similar time period. Workloads driven by a variety of users co-existing with varied business needs may not satisfy this assumption in a Cloud environment. Users are characterised with unique patterns of job submissions and moreover, a single user might also submit different types of jobs in a single session. Thus modelling the incoming job trend as a simple auto regression within an observation period may not scale well under co-existing users with varied characteristics. RPPS [15], a prediction framework based on simple ARIMA technique, has been proposed to predict the future workload trends. The resource usage pattern has been fed as a time-series into the predictor to predict the workload pattern for a short-term time. ARIMA model is subjected to confidence limit bounds to the actual forecast determining the over and under-estimation errors of the forecast. Since the Cloud workloads exhibit extreme dynamism in both the arrival frequency and the number of expected submissions, a simple ARIMA forecast may not deliver a precise forecast of the workload patterns in Cloud environments. The larger variance of the workload pattern resulting from the fluctuating job submission trend of the users causes increased residuals in the training data which can significantly affect the prediction accuracy. A second order autoregressive method [32] is deployed to predict the workload patterns in Cloud environments for an effective resource management also suffers similar drawbacks.

The dynamic nature of the user behaviours and the workload behavioural patterns impose various levels of challenges in developing an effective prediction model. Since periodicity is an important metric determining the prediction accuracy, incorporating the measure of periodicity and their influence over the prediction accuracy is essential for an efficient prediction framework. However existing works of predicting the incoming job trends for resource scaling have not given suffice emphasis to the inherent degree of periodicity among the Cloud users and their corresponding workloads. Furthermore, the degree of inherent periodicity highly fluctuates across different workloads and users which necessitates treating every jobs submitted by every users uniquely during an observation period. Treating all the incoming workloads in a common way during a given session might not help precisely understanding the characteristics of the incoming workloads for further predicting their future trend. It is commonly evident that most of the existing prediction techniques utilise historical data for estimating the future trend. However, the correlation between the historical samples and the currently interested observations have not been essentially validated, choosing the most appropriate historical samples is very important in determining the prediction accuracy. With this in mind, this paper proposes InOt-RePCoN (Influential Outlier Restrained Prediction with Confidence Optimisation), for the purpose of predicting the user behaviours in terms of the anticipated number of job submissions, session duration and the job submission interval. User behavioural pattern have been given special emphasis in our prediction model to predict their corresponding job submission trend, since users are the actual drivers of the workloads. Every jobs submitted by every single users during an observation period have been uniquely treated to combat user diversity. Periodical behaviours of the Cloud users and their corresponding workloads are analysed in relation to the business hours of the datacentre operation for incorporating the knowledge of inherent periodicity in the prediction model. Exploiting periodicity by considering the two important periodical effects such as time-of-the-day and the day-of-the-week effects, our proposed model predicts the user behavioural trend based on a confidence optimised ARIMA model utilising the most appropriate historical data samples. The selection of the historical samples for optimising the prediction confidence has been validated by the measure of correlation between the historical samples and the current observation trend, which ensures reliable level of prediction accuracy.

3. Background

3.1 Cloud Workloads

A typical Cloud workload [33] arrives at the Cloud datacentre in the form of jobs submitted by the users. Every job includes certain self-defining attributes such as the submission time, user identity and resource requirements in terms of CPU, memory and disk space. A single job may contain one or more tasks, which are scheduled for processing at the Cloud servers. A single task may have one or more process requirements. Tasks [34] may have varied service requirements and characteristics such as throughput, latency, jitter, etc., even though they belong to the same job. Two jobs with the same resource requirements may not be similar in their actual resource utilisation levels because of the variations found among the tasks contained within the jobs. Service providers generally record the resource utilisation levels of every scheduled task and maintains the user profiles. The attributes encompassed by the Cloud workloads such as job name, user name, submission time, resource requests

and usage pattern, etc., can be exploited to derive the behaviours of both the Cloud users and their corresponding workloads.

3.2 Data Sample

This research work explores the Cloud trace logs [35] released by Google, featuring more than 650000 jobs over 28 days of datacentre execution. The studied datacentre includes a total of 12500 servers and the system uses Linux Containers (LXC) for resource isolation and virtualisation and runs multiple isolated processes within a single server. Each task runs within its own container and services are provided to a multitude of applications. The event analysis of the trace logs based on our previous research [12] has been presented in Table. 1. The trace log data has been sampled on a daily basis with a single day spanning across 24 hours starting from 12.00 am for a given day. In order to accurately model the time-of-the-day and day-of-the-week behaviours of the workloads and the users, the trace log data has been sampled in such a way that the trace time starts exactly at 12.00 am on a Sunday.

Table 1. Trace Log Statistics

Number of Days	28
Total Number of Job Submissions	650892
Total Number of Task Submissions	46093201
Number of Operating Servers	12500
Average Number of Users per Day	190

3.3 Characterising Predictability

Both the Cloud workloads and the users usually exhibit temporal and spatial correlations driven by repeatable business behaviours, which generates a periodical pattern for characterising users and their corresponding job submissions. For instance, generating weather reports is a typical example of a timely recurring job submission. This section is aimed at uncovering the predictability features of both the Cloud users and their workloads.

3.3.1 Cloud Users

In a Cloud datacentre, job submissions are usually characterised by associated user ID, assigned logical name and the corresponding resource requirements. Jobs submitted under the same user name implies that all such jobs are submitted by a single user. This user name is a randomly allocated string and are uniquely assigned to the users. A single user may also have various user profiles under different user IDs, which would lead to the generation of various user driven profiles for a single user. Such user profiles might exhibit similar user behavioural patterns since such multiple user profiles belong to a single user. Despite this inherent similarity among the user profiles, matching the ownerships of the jobs submitted under different user profiles is often tedious. But the jobs characterised with similar behavioural patterns closely correlated with common user profiles of similar characteristics can be treated in a common way for predictive analytics. Furthermore, the active number of concurrent users in an execution session is another important factor that affects the prediction accuracy. Users co-existing in a service session not necessarily have similar resource requirements. Usually, Cloud providers employ a higher level of parallelism under an increased number of concurrent users requesting similar resources. User profiles are very dynamic in a way that every user has a potential access pattern and do not have a static IP address as it is dynamically assigned to the users from a limited number of address pools. This often leads to the assignment of the same IP address to several users. User behaviours evolve over time, and thus the snapshots of user profiles obtained over a relatively shorter period are mostly imprecise.

3.3.2 Cloud Workloads

Cloud workloads are governed by various intrinsic attributes from which their predictability can be extracted. Job execution duration and number of tasks within jobs are the two important metrics defining workload characteristics. Job execution duration [36] is usually bimodal, with tasks contained within the jobs either running for a shorter time or a longer time. Long running tasks can be further classified as user facing tasks and compute intensive tasks. The former runs continuously with quicker user interactions and the latter generally refers to the processing of the weblogs. Shorter duration tasks can be further classified as highly parallel user requests of both CPU and memory resources, shorter CPU and shorter memory intensive tasks respectively. Majority of the Cloud jobs run for less than 15 minutes [37] and a very few number of jobs are more than 300 minutes in duration, with the duration of latency sensitive jobs being less than 30 minutes on average. Task duration heavily depends on the nature of the user behaviours and their interactions. Generally, a single job may contain tasks of both shorter

and/or longer durations, and tasks running longer usually consume most of the allocated resources. It is worthy of note that jobs are generally governed by various constraints such as specified server and scheduling requirements. An efficient Cloud infrastructure effectively manages such constraints, still a few type of jobs can encompass more than 400 constraints impacting the execution duration. Most of the jobs in a typical Cloud datacentres encompass smaller to medium number (100 on average) of tasks, and a very few number of jobs have a single task. On the contrary, a very few jobs may also contain more than 2000 tasks. Thus majority of the Cloud users submit jobs with smaller number of tasks and a very few users submit jobs with larger proportion of tasks. In the case of jobs submitted with multiple tasks, the execution duration of the entire job is actually the summation of all the task duration contained within the corresponding job. Both the smaller number of jobs with increased number of tasks and the larger number of jobs with fewer tasks have distinctive impacts on the overall datacentre behaviour.

3.4 Cloud Dynamicity

Though the Cloud workloads show predictable properties, the heterogeneity found among both the Cloud users and the workloads [38] impose several challenges in predicting their future behaviours. With the Cloud server resources exhibiting heterogeneity among their operating conditions, process capabilities etc., dynamism [39] is also evident among the workloads in terms of their arrival frequency, resource request and utilisation levels. Such a dynamic nature of the datacentre process environments impose various levels of challenges in carrying out a real-time prediction analytics for driving effective decision making. The complexities in decision making driven by prediction analytics can be demonstrated in two different scenarios, as shown in Fig. 1. Firstly, job submissions varying in accordance with a periodic oscillation of constant amplitude. In such scenarios, predicting either the peeks or the valleys can be utilised to manage the datacentre resources by the way of scaling up/down the active resources at an optimum level based on the curve trend. Secondly, the job submissions varying abruptly with the submission curve characterised by uneven amplitudes and very close occurrences of peaks and valleys. Given the two scenarios, the former allows reasonable time interval where a much better datacentre management can be achieved. But the later allows a very lesser time scale to carry out the predictive analytics and further datacentre management driven by the prediction.

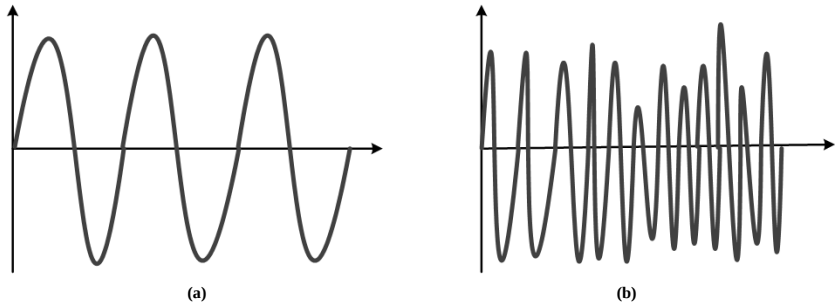


Fig. 1. Cloud dynamism (a) Constant workload (b) Fluctuating workload

Another challenge imposed by the workload behaviour is the job submissions from brand new users. Such newly arriving jobs may or may not have a pre-existing user or job profiles to which they can fit into. If they don't fit into an existing profile, then new user and job behavioural profiles should be created for every new users. Cloud workloads may also contain anomalies [40] and jobs submissions from malicious users. Such anomalies generally exhibit an abnormal behavioural pattern. Some of the runtime factors such as user access patterns, user concurrency, and resource usages often result in contextual anomalies which are unavoidable in Cloud environments. It is possible that these anomalies could also be categorised as newly arriving jobs submitted by brand new users. Conversely genuine workloads might also be classified as anomalies, which would result in a higher number of false positives. Such a classification leads to unpleasant events such as wrong prediction, further causing service outages and execution failures.

3.5 User Profile Definition

User requests arrive in the form of job and task requirements at the Cloud datacentres. Job Profile is a composite consisting of Submission time t_s, user (name) n_u submitting the job and the logical job name n_j, as shown in (1).

Task profile is a composite consisting of submission time t_s, user name n_u and the corresponding resource request $r_{(c,m)}$, as shown in (2).

$$J = \{t_s, n_u, n_j\} \tag{1}$$

$$T = \{t_s, n_u, r_{(c,m)}\} \tag{2}$$

Submission time and the user name are the commonly identified metrics among the job and task profiles and a combination of job and task profiles are used to build the user profiles. Usually a single user can trigger multiple job requests in a datacentre environment. All those logical jobs can have one to several number of tasks each, with all those tasks belonging to that corresponding user. In a typical Cloud datacentre, different task executions of the same job type will have a common logical name. Thus, the individual profiles for every users are defined by the way of incorporating their respective predictive parameters for the purpose of exploring their predictability. A Cloud user profile can thus be defined as a composite *U*, consisting of the time of job submission t_s, and user name n_u, job name n_j and the associated resource demands in terms of CPU and memory $r_{(c,m)}$, as shown in (3).

$$U = \{t_s, n_u, n_j, r_{(c,m)}\} \tag{3}$$

4. InOt-RePCoN Framework

Our proposed prediction model is aimed at predicting the user behavioural trend in terms of their anticipated number of job submissions, session duration and the job submission interval. This section describes the integral components of our proposed prediction model InOt-RePCoN, as shown in Fig. 2. InOt-RePCoN encompasses three integrated components such as a Rule Miner, a Validator and a Predictor.

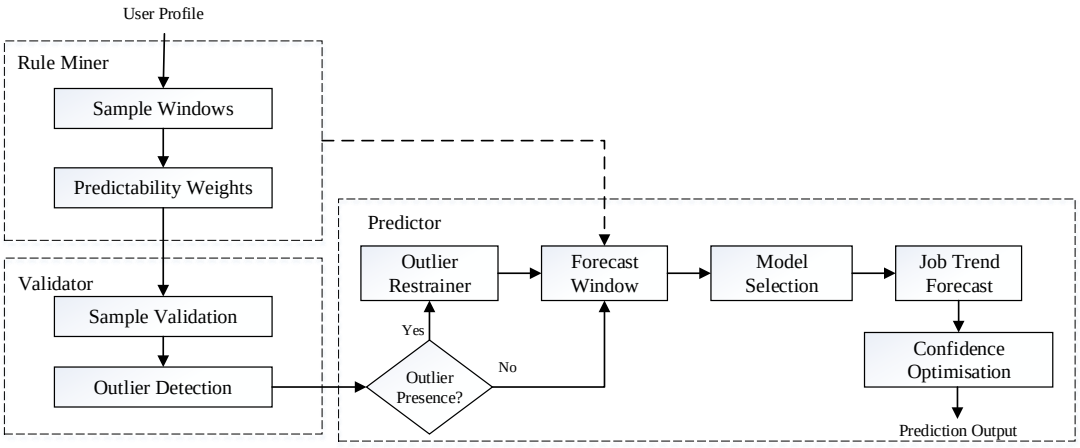


Fig. 2. InOt-RePCoN framework

The integrated components of our proposed prediction model will have the following functionalities.

- *Rule Miner*: The main functionalities of the rule miner are to select the historical samples for training the prediction model and to compute the predictability weights for the target users and their jobs. The rule miner initially reads the user profile from the incoming user request. By the way of delving into the time-of-the-day and day-of-the-week effects, the rule miner selects the historical samples based on the day and time of the current sampling period. The chosen historical samples are of same duration to the current sample. Based on the two aforementioned periodicity effects, the rule miner computes a predictability weight for the user and their workloads comprised in the current sampling period.
- *Validator*: The validator incorporates two sub-components such as the sample validator and an outlier detector. The sample validator chooses the most suitable historical sample from those samples initially chosen by the rule miner. This historical sample is chosen based on the measure of the degree of correlation between the current sample and the historical samples. Further to this, another sample from the historical traces is chosen called the reference sample which is the next successive set of sample to

the one chosen by the validator. The outlier detector measures the degree of residuals present in the prediction sample and in the chosen historical samples.

- *Predictor:* The main functionalities of the predictor are to forecast the number of submissions, session duration and submission interval trend for the target users and jobs based on the samples chosen by the validator. It incorporates five sub-components such as an outlier restrainer, window forecaster, prediction model selector, predictor and a confidence optimiser. Outlier restrainer restrains the effects of influential outliers present in the prediction sample up on the prediction accuracy. Window forecaster computes the number of submissions and session duration anticipated for the target users and jobs during an observation period. The model selector and predictor are modelled to forecast the anticipated future trend of job submission interval for the target users by selecting the most appropriate predictor values. Further the accuracy of the forecast is enhanced by optimising the confidence interval of the forecast.

Detailed descriptions of the integrated components of our proposed prediction model is presented in the next section.

5. Prediction Mechanism

5.1 Rule Miner

The rule miner receives the input consisting of the current sample of user trend and has two important functionalities. Firstly, the rule miner selects the prediction samples from the historical data, by the way of matching the start-end time and duration of the current sample such that the chosen historical samples are identical in duration and start-end time of the current sample. Two such historical samples are selected, one from the same representative day in the previous week of the current sample in order to validate dual effects of time-of-the-day and day-of-the-week effects collectively. The second historical sample is chosen from the previous day of the current sample to validate the time-of-the-day effects. After choosing the samples, rule miner forms four different sample windows such as the current sample window (W_c) containing the current user trend from which the future trend is expected to be forecasted in the prediction window (W_p), window 1 (W_1) is built with the dual effect sample, and window 2 (W_2) for the time-of-the-day sample accordingly. Fig. 3 illustrates the various sampling windows formed by the rule miner, in reference to a randomly chosen Day 10, Wednesday, 9 am – 10 am data contained in the current sample. Secondly, the rule miner computes a predictability weight P_s for every users contained in W_c, along with assigning predictability weights for all the type of jobs submitted by the target users. This predictability weight determines the degree of predictability of the users and jobs depending on the current trend of users and jobs satisfying the sample window rules of the rule miner.

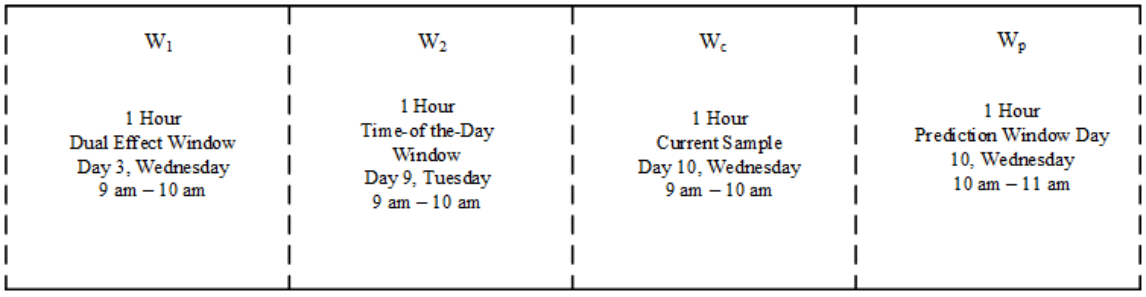


Fig. 3. Rule miner window illustration

The predictability weight, P_s is assigned to every users and every job types belonging to the users by the measure of the W_c samples satisfying the day-of-the-week and time-of-the-day effects in accordance with W_1 and W_2. The rule miner assigns four levels of prediction weights to the users and their jobs. A weight of level 3 is assigned to the users and jobs satisfying both the dual effect window W_1 and the time-of-the-day window W_2. A weight of level 2 is assigned to users satisfying only the dual effect window W_1. A prediction weight of level 1 is assigned to users satisfying only the time-of-the-day window W_2. User not satisfying any of the two windows will be assigned with a predictability weight of level 0. Thus level 3 implies a higher degree of predictability through to level 0 implies a poor degree of predictability for users and jobs. By assigning predictability weights to the users, InOt-RePCoN exploits the periodicity among the user behavioural pattern in terms of their submission trend during the two historic sample windows for choosing the most appropriate sample for prediction. The predictability weights are used to determine the error margin and forecast accuracy for users and jobs in our

prediction model. Higher the prediction weight better is the expected correlation of the predicted trend with the actual trend of the corresponding users and jobs. In other words, an increased level of predictability weight reflects the increased expectation of prediction accuracy. After computing the predictability weight for every users in the current sample window, the rule miner constructs a predictability weight table based on the list of users l_c, l_1 and l_2 respectively contained in the current sample window, window 1, and window 2. These process are repeated for every jobs submitted by every users contained in the current sample window in order to assign the predictability weights to all the jobs submitted by every users. The construction of the predictability weight table based on the predictability weight computation for users and jobs are detailed in section 6.3.

5.2 Validator

After assigning the predictability weights to both the users and jobs in the current sample window, our proposed model further validates the similarities of the user behavioural trend in the current sample window with both window 1 and window 2. Though the rule miner relies on both the time-of-the-day and day-of-the-week effects to assign the predictability weight, this similarity measure is conducted for the purpose of training the most suitable historical sample to the predictor from W_1 and W_2. Every single users and their jobs are analysed uniquely, since users co-existing in a given session usually exhibit varied job submission trend and service requirements. Thus the behavioural trend of the users in terms of their job submission patterns in the three sample windows are analysed to measure the similarity of the user behaviours in the current sample window with both the two historical windows.

5.2.1 Similarity Analysis

A single user might submit several jobs and thus characterised by multiple job submission trends. Thus the validator analyses the similarity measure for every individual jobs submitted by the users. For this reason, our proposed model is aimed at forecasting the user behavioural trend for individual jobs submitted by the corresponding users. Firstly, the submission interval S_i for a given job is calculated in the three sample windows using (4).

$$S_i = t_{j(i+1)} - t_{ji} \tag{4}$$

where, t_{ji} is the submission time of job j at time i, and $t_{j(i+1)}$ is the submission time of the job j at time i+1, which is the next successive submission time of job j.

The validator measures the degree of correlation among the submission intervals of job j by validating the linear dependence of every consecutive submissions of job j in the current sample window with those in window 1 and window 2 respectively. The linear dependence in the submission interval of job j is validated by the measure of the correlation coefficient by modelling S_i of job j as a time series. Since the Cloud workloads are dynamic in nature, submission interval of the same job by the same user within a single sample window might exhibit significant variation resulting in the presence of outliers in the job submission interval. An increased presence of such outliers cause the submission interval pattern of the corresponding jobs to exhibit a non-linear trend and significantly affects the prediction accuracy. More the presence of outliers higher is the deviation of the observation from a linearity trend. Thus, the validator measures the presence of outliers in the submission interval trend of the target jobs from users for the purpose of measuring the degree of linearity in the job submission interval. Presence of outliers is quite common in the job submission trend owing to the increased dynamicity in Cloud environments. The characteristics of an outliers among the data points can be defined as in (5).

$$O_t = \begin{cases} 0 & if\ |r_i| \leq C(p) \\ 1 & otherwise \end{cases} \tag{5}$$

where, O_t is the outlier, r_i is the residuals with $i = 1, \ldots, n$ and $C(p)$ is the cut-off value for deciding the residuals. A simpler correlation coefficient for similarity measure do not scale well and might result in an incorrect estimation of the dependency for validating linearity under the presence of outliers. Based on the degree of the outliers, the validator decides whether the prediction sample in the current sample window requires subjecting to outlier restraining before training into the predictor (further detailed in section 5.3.2, 6.4 and 6.7). Now, the validator computes the confidence and prediction ellipses for the job submission trend of the target jobs from target users contained in the current sample window and window 1 and window 2 respectively for the purpose of accurately estimating the correlation coefficient and the presence of outliers among the submission trend.

A bivariate distribution has been adopted for analysing the Job submission trends, the validator contrasts the job submission trend in the current sample window against those both in window 1 and window 2 but one at a time. Confidence ellipse defines the event population mean and the prediction ellipse defines the confidence bounds of predicting the future observations. The Confidence and Prediction ellipse computations are defined as follows. Let *Z* and *S* be the sample mean and covariance matrix of a random sample of size n with mean μ and covariance $\sum$. The variable $Z - \mu$ is a bivariate distribution with zero mean and covariance of $(1/n)\sum$, and it is independent of *S*. From Hotelling's T^2 statistic, we have (6).

$$T^2 = n(Z - \mu)' S^{-1} (Z - \mu) \tag{6}$$

Now a 100(1-α) % confidence ellipse for μ is computed using (7), where $F_{2,n-2}(1-\alpha)$ is the $(1-\alpha)$ critical value of a F distribution with degrees of freedom 2 and $n-2$.

$$\frac{n}{n-1}(Z - \mu)' S^{-1}(Z - \mu) = \frac{2}{n-2} F_{2,n-2}(1-\alpha) \tag{7}$$

The prediction ellipse estimates the new observations Z_n as a bivariate normal variate with zero mean and covariance $(1 + \frac{1}{n})\sum$, independent of S, given by (8).

$$Z_n - Z = (Z_n - \mu) - (Z - \mu) \tag{8}$$

Now a 100(1-α) % prediction ellipse is given by (9).

$$\frac{n}{n-1}(Z - \mu)' S^{-1}(Z - \mu) = \frac{2(n+1)}{n-2} F_{2,n-2}(1-\alpha) \tag{9}$$

Both the generated confidence and prediction ellipses will have common centre (the sample mean), common major and minor axis. The degree of association between the consecutive job submissions and their submission interval is measured using Pearson correlation coefficients as shown in (10). A positive correlation coefficient insists a close correlation between the two variables x and y. This correlation coefficient measures the degree of linear dependency between consecutive submission of jobs and their submission interval. By the way of generating the confidence and prediction ellipses, the validator presents the residuals in the job submission trend falling beyond the estimated ellipses. Since outliers directly affect the prediction error margin, presence of such residuals cannot be ignored whist generating the prediction ellipses.

$$\rho_{xy} = \frac{Cov(x,y)}{\sqrt{V(x)V(y)}} = \frac{E((x-E(x))(y-E(y)))}{\sqrt{E(x-E(x))^2 E(y-E(y))^2}} \tag{10}$$

By generating independent prediction ellipses for the trend of the target jobs and users in the current sample window, window 1 and window 2, the presence of residuals and the correlation coefficient is validated between the three sample windows for the purpose of training the most suitable historical sample into the predictor. The correlation coefficient is determined by various intrinsic factors such as the time, user intention, business pattern etc. For instance, the current trend of the target users and jobs might have a close correlation with the time-of-the-day effects or with the day-of-the-week effects or both. Though window 1 satisfies both the time-of-the-day and day-of-the-week effects, window 2 will still comprise the most recent historical sample (just a day old), with window 2 comprising a week old sample. Thus validating the correlation coefficient for similarity measure between the current and historical samples is crucial in determining the prediction accuracy.

5.3 Predictor

This sections details our proposed prediction framework based on autoregressive integrated moving average (ARIMA) technique integrated with outlier restrained confidence optimisation. Auto Regression scales well for prediction when the prediction samples characterise an inherent periodicity [12]. Job submission trends of the users exhibit periodicity among the parameters such as the submission time, and submission frequency deciding the submission interval between every consecutive submissions. The predictor models the job submission trend of the users as a time series to extract the periodical predictive characteristics from the current job submission trend of the users. The integral components of the predictor are described along with their functionalities as follows.

5.3.1 Stationarity Test

Initially, the predictor conducts a stationarity test upon the submission trend of the target jobs and users in the current sample window for testing the degree of stationarity in the time series of the job submission time and the submission interval. The stationarity of the predictive sample are evaluated using an Augmented Dickey-Fuller (ADF) t-statistic test for stationarity by subjecting the submission trend for null-hypothesis. The degree of stationarity characteristics of the job submission time and the submission interval are used to validate and select the appropriate ARIMA model for the purpose of accurately forecasting the future observations. With the job submission behaviour of the users following a continuous time-series trend and expected to have a slow-turn around the data points, the ADF test is conducted using (11).

$$\Delta z_t = \alpha_0 + \Theta Z_{t-1} + \gamma t + \alpha_1 \Delta Z_{t-1} + \alpha_2 \Delta Z_{t-2} + \ldots + \alpha_p \Delta Z_{t-p} + \alpha_t \quad (11)$$

The t-statistic on the Θ coefficient is used to evaluate the degree of stationarity and the submission trend is differenced when the trend exhibits non-stationarity. More the t-statistic is negative, more is the data points are in trend. The null hypothesis of the ADF t-statistic test is given by,

$$H_o: \Theta \begin{cases} = 0 \; for \; data \; needs \; to \; be \; differenced \\ < \; 0 \; for \; the \; data \; trend \; is \; stationary \end{cases}$$

5.3.2 Outlier Suppression

Prior to training the input submission trend into the predictor, the prediction sample of the current trend of job submission of the target users and jobs is subjected to robust regression for the purpose of restraining the effect of the influential outliers. The presence of the outliers in the submission trend is estimated based on (5), and the sample is subjected to robust regression depending on the degree of the presence of the outliers. Prediction samples suffering marginal or no outliers may not require robust regression. But, job submission trends of the users usually suffer increased variance within an observed time period. Thus outliers and residuals are quite prominent in the trend of user submission behaviours in Cloud environments. The presence of such outliers in the prediction sample increases the error margin and often results in inaccurate prediction results. Thus it is essential to supress the influence of the presence of outliers for the purpose of achieving reliable level of prediction accuracy. Usually the influence of the presence of outliers are dominant in the Y plane of the job submission trend of the users since the contamination of the data points resulting from the variances are mainly witnessed in the response direction. Thus the predictor initially estimates the presence of outliers by the degree of the variance in the submission interval, and further supress the presence of such outliers by subjecting the prediction sample with robust regression as shown in (12) to (13). The estimation of outliers and suppression is illustrated in section 6.4 and 6.7 respectively. With the data contamination being witnessed in the response direction, robust regression algorithm computes the M estimates for regression based on iteratively reweighted least squares (IRLS). An IRLS fit is carried out in every iteration based on a set of weights applied depending on the presence of residuals until convergence is achieved. The M estimator $\Theta_M \; of \; \Theta$ minimises the sum of less rapidly increasing residual functions under residuals r_i.

$$Q(\Theta) = \sum_{i=1}^{n} \rho\left(\frac{r_i}{\sigma}\right) \quad (12)$$

where ρ is the square function $\rho(z) = z^2$. If σ is known, then Θ_M is the solution for the system of ρ equations by the derivatives with respect to Θ.

$$\sum_{i=1}^{n} \Psi\left(\frac{r_i}{\sigma}\right) x_{ij} = 0, j = 1 \ldots, p \quad (13)$$

where $\Psi = \frac{\partial \rho}{\partial z}$, and the weight function depending on the residuals is $w(z) = \frac{\Psi(z)}{z}$. Robust regression is proceeded by alternately improving Θ in a location step and σ in a scale step, until convergence is achieved, whenever there is a relative change in the scaled residuals for the purpose of restraining the effect of influential outliers.

5.3.3 Forecasting Window

After supressing the effect of the influential outliers in the prediction sample, it is essential to initially forecast the characteristic number of job submissions and session duration for the target users and jobs anticipated in the prediction window W_p. Session duration is the duration between the first and the last submission of the jobs from the users during the period of observation. Now, another sample window named reference window W_r is

introduced comprising historical samples from the next successive observation period of the historical window W_1 or W_2, whichever is finally validated by the sample validator. The length of this reference window W_r is equivalent to those of the four sample windows initially constructed by the rule miner. After the construction of the reference window, the predictor generates a relative error margin E_1 for both the anticipated number of job submissions and the session duration for the target users and jobs contained in W_c in reference to the finally validated sample W_1 or W_2, using (14) (W_1 is considered as the validated window for the below descriptions).

$$E_1 = \frac{|V_{1(n,s)} - V_{c(n,s)}|}{V_{c(n,s)}} * 100 \tag{14}$$

where, V_{cn} and V_{1n} are the number of actual job submissions observed in W_c and W_1 respectively, and V_{cs} and V_{1s} are the session duration counter-part. In addition to E_1, another expected relative error margin E_2 is computed based on the previously computed predictability weight P_s for the target users and jobs, using (15).

$$E_2 = \frac{V_{r(n,s)}}{100} * e_p \tag{15}$$

where, e_p is expected error percentage based on the predictability weights of the job submission trend of the user behaviours, set as 10, 15, 20 and 25 respectively for P_s values of 3, 2, 1 and 0, and $V_{r(n,s)}$ is the observed number of submissions and session duration for the target jobs and users in the reference window W_r. The final error margin E is computed for the forecasting window in terms of the anticipated number of job submissions and session duration as in (16). The error percentage is computed separately for the number of submissions and the session duration accordingly.

$$E = E_1 + E_2 \tag{16}$$

Now the anticipated number of submissions V_{pn} and session duration V_{ps} for the target users and jobs is computed for the forecasting window W_p, as $V_p = \{V_{pn}, V_{ps}\}$ with optimised error margin, using (17).

$$V_p = \begin{cases} V_r & for\ V_w = V_c \\ V_r - E\ of\ V_r & for\ V_w \ll V_c \\ V_r + E\ of\ V_r & for\ V_w \gg V_c \end{cases} \tag{17}$$

where, V_w is the values of the actual number of job submissions and the session duration for the target users and jobs observed in the historical window (W_1 or W_2) validated by the sample validator. In (17), $V_w \ll V_c$ insists significant difference between the values in $W_{(1\ or\ 2)}$ and W_c, and a value is concluded to significantly different if the difference is greater than half of E. If the difference is not significant, the values are considered to be equivalent.

5.3.4 Model selection and ARIMA Forecast

A random variable of a stationary time series has statistical properties over time with constant amplitude around the mean. But a non-stationary trend of a time series shows a more fluctuating amplitude and variance around its mean. A random variable of such series with non-stationarity usually incurs a combination of signal and noise. Though regression assumes a model for forecasting the future trend, it is essential to select the most suitable regression model with appropriate subset of predictor variables based on the trend of the variables contained in the prediction sample. Based on the regression estimates on trend of the prediction sample, the predictor selects the appropriate ARIMA model for predicting the future trend of job submission interval. From the initial ARIMA identification for stationarity, the predictor also identifies the degree of Auto Regression and Moving Average processes required to be optimised for autocorrelations existing in the original series in the case of stationarity or in the differenced series in the case of non-stationarity in the data points, by the measure of the ACF and PACF functions. In a stationarised series, AR signatures associates a positive correlation to act as a partial difference in the forecasting equation whereas MA signatures associates a negative correlation to partially cancel the order of differencing in the forecasting equation. Thus in a stationarised series after differencing the original time series, an AR signature mimic the first difference and an MA term moderate the first difference, with a redundant AR-MA pair cancelling out the effects of each other. In general, ARIMA model delivers a forecast $\hat{y}_t$ for a stationary or a differenced time series, in which the predictors consist the lags of the dependent variable or the forecasting errors. A non-seasonal ARIMA model can be classified as ARIMA (p, d, q), where p is the autoregressive term, d is the number of non-seasonal differences required for stationarity, and q is the number of lagged forecast errors in the prediction equation. An ARIMA (p,d,q) with y denoting the d^{th} difference of Y, can be descried in (18).

$$y_t = \begin{cases} Y_t, & if\ d = 0 \\ Y_t - Y_{t-1}, & if\ d = 1 \\ Y_t - 2Y_{t-1} + Y_{t-2}, & if\ d = 2 \end{cases}$$

$$\hat{y}_t = \mu + \phi_1 y_{t-1} + \ldots + \phi_p y_{t-p} - \theta_1 e_{t-1} - \ldots - \theta_q e_{t-q} \quad (18)$$

This ARIMA forecast is expected to deliver the submission interval of consecutive submissions for the target users and jobs in the current sample window, with an upper and lower bound 95% confidence interval set by the predictors.

5.3.5 Confidence Interval Optimization

In order to improve the prediction accuracy of the ARIMA forecast and to reduce the bound limits of the 95% confidence window delivered by the ARIMA predictors, our proposed model further optimises the ARIMA forecast with our novel confidence interval optimiser. Since a continuous trend is expected for the job submissions, submission interval between two consecutive submissions can only be a positive value in time. In order to satisfy this positive bound requirements of the confidence window, the optimiser nullifies the effect of the negative lower bounds against the corresponding positive upper bounds of the ARIMA confidence window to obtain the optimised window limits, as shown in (19).

$$L_m = \begin{cases} L_{li}\ , & for\ L_{li}\ significantly\ positive\ (i = 1, \ldots, n) \\ U_{li} + L_{li} & for\ L_{li}\ significantly\ negative\ (i = 1, \ldots, n) \end{cases} \quad (19)$$

where L_m is the mean bound limits after nullifying the effects of the negative lower bounds of the ARIMA confidence interval and n is the number of submissions initially set by the window forecaster. Now the predictor further optimises the confidence interval against the actual forecast of the ARIMA model, by assuming a zero mean for the lower bound limits. With a zero mean for lower bounds in the confidence interval, the upper bound limits L_o are optimised based on the submission interval of the ARIMA forecast I_f, and the actual submission interval I_r observed in the reference window W_r, and the computed mean limit L_m, using (20).

$$L_o = \begin{cases} I_f, & for\ I_r\ not\ available \\ L_m + I_f, & for\ I_f < I_r \\ L_m - I_f, & for\ I_f > I_r \end{cases} \quad (20)$$

Thus our proposed model predicts the job submission interval based on the forecasting window along with optimising the confidence interval of the ARIMA forecast. This optimised confidence interval reduces the interval bounds of the ARIMA confidence window for better accuracy and reliability in the prediction output. Usually in a Cloud Computing environment, under-prediction would have more disastrous effect on energy consumptions than an over-prediction. While the former results in additional wait-time for the service providers, the later might lead to a quicker arrival of the anticipated job than expected. When jobs arrive quicker than expected, services can still be availed with a marginalised wait time for the users. But jobs arriving later than expected might result in early provisioning of the resources causing undesirable energy expenditures. Thus InOt-RePCoN is aimed at reducing the probabilities of under-predictions of the job submission interval in the final optimised confidence interval of the forecast. Our proposed model further optimises the bound limits to deliver the confidence interval W_{con}, for the purpose of reducing the probabilities of under-prediction using (21). Most often, Cloud user behaviour trend of job submission interval is governed by the presence of influential outlier. In this case, the anticipated trend is expected within the bounds of W_{con} and the zero mean lower bounds of the forecast. For samples with no or minimal influence of the outliers, the future trend is expected to be in correlation with W_{con}.

$$W_{con} = \begin{cases} I_f & for\ \frac{I_f}{2} > L_o > (I_f * 1.5) \\ L_o & otherwise \end{cases} \quad (21)$$

6. Model Validation

Model validation is the process of substantiating a computerised model to determine whether its applicability possess an acceptable range of accuracy and reliability consistent with the intention of the model application. This section validates our proposed prediction model by the way of training a real-life Cloud datasets into the model for forecasting the user behaviours.

6.1 Data Preparation

The dataset comprises the job and task profiles across a period of 28 days of datacentre execution. The entire dataset is prepared with a day-wise sampling, with a single day spanning across 24 hours starting from 12.00 am for a given day. Then, Day 10 Wednesday is randomly as our test sample and InOt-RePCoN is expected to predict the user behavioural trend during the period of 1 am-2 am, using the sample of 12 am- 1 am as the current window W_c sample. Day 10 is further split into 24 samples, each spanning across a period of our hour so that all the sample windows of the rule miner is one hour long. The same process is repeated on Day 9 Tuesday and Day 3 Wednesday for validating samples for Window 1 and Window 2 accordingly.

6.2 Sample Selection

After processing the raw datasets for our analysis, input the samples are trained into InOt-RePCoN. Based on the prediction objective of forecasting the user behaviours during 1 am – 2 am on Day 10, Wednesday, the rule miner constructs the sample windows. Now, the current sample window (W_c) comprises the data from 12 am – 1 am, Day 10 Wednesday, window 1 (W_1) comprises the data sample from 12.00 am – 1.00 am, Day 3 Wednesday and Window 2 (W_2) comprises the data sample from 12.00 am – 1.00 am, Day 9 Tuesday. Our proposed prediction model is expected to predict the user behaviours for the predictor window W_p, which is 1 am – 2 am, Day 10 Wednesday. Further to validate and optimise the confidence interval of the forecast results and to set the forecast window, the rule miner further samples the data obtained from 1 am - 2 am, Day 9 Tuesday and 1 am – 2 am Day 3 Wednesday for the purpose of validating the reference window (W_r).

6.3 Predictability Weight Computation

The current sample window W_c comprises a total of 55 users, implying all those 55 users have co-existed with their characteristic job submissions during the one hour observation period of W_c. The rule miner builds the historic sample windows as described earlier, and computes the predictability weight for all the 55 users. For the ease of reading, the 55 users of W_c are named as User 1 through to User 55, in the descending order of their corresponding number of job submissions. Table 2 presents the submission observations for the first 10 users from W_c. Users characterised with increased number of job submission leave sufficient behavioural traces from which their behavioural periodicity can be extracted.

Table 2 User submission statistics from Wc

User Name	Number of Job Submission	Event Proportion (%)
User 1	361	22.59
User 2	339	21.21
User 3	153	9.57
User 4	127	7.94
User 5	75	4.69
User 6	56	3.50
User 7	36	2.25
User 8	36	2.25
User 9	34	2.12
User 10	32	2.002

Now, the rule miner computes the predictability weight for all the 55 users, as shown in Table 3. Out of the total 55 users, 37 users are exhibiting a level 3 predictability weight, reflecting their increased predictability. Following level 3 predictability, 4 users are assigned with level 2 predictability weight, 4 users with level 1 predictability, and 10 users with level 0 predictability weight respectively. These 10 users with level 0 predictability weight exhibit a very low probability of accurate prediction, as they do not fit into an existing trend. This could be because of the limited user profiles or because they might either be brand new uses or possible anomalies. The prediction accuracy for such brand new users will be enhanced after obtaining sufficient number of behavioural traces from which their user profiles can be built and recorded.

Table 3. User Predictability Weights

Users ($\boldsymbol{n_u}$)	User Predictability Weight ($\boldsymbol{P_{su}}$)
46, 47, 16, 12, 34, 39, 26, 4, 48, 49, 17, 40, 9, 50, 5, 3, 27, 52, 53, 41, 25, 10, 32, 54, 31, 21, 7, 29, 33, 22, 38, 20, 18, 43, 11, 1, 55	Level 3
30, 28, 42, 13	Level 2
23, 14, 51, 36	Level 1

44, 45, 15, 2, 6, 35, 24, 37, 19, 8	Level 0

After assigning predictability weights to the users, the rule miner individually explores the jobs submitted by all the 55 users contained in $W_{c.}$ For space limitations, only the computed predictability weights for all the jobs submitted by User 1 are presented in Table 4. User 1 has submitted the maximum number of jobs in $W_{c.}$ Despite User 1 exhibiting a higher level of predictability, not necessarily all the jobs submitted by User 1 should exhibit a predictability weight of level 3. User 1 has submitted 17 different type of jobs across 361 submissions in W_c, as shown in Table 4. For the ease of readability, jobs submitted by User 1 are named as Job 1 through to Job 17, presented in the descending order of the number of submissions. Now, the rule miner constructs the predictability weight table for all the jobs submitted by User 1. Table 4 presents the predictability weight table comprising all the jobs submitted by User 1, along with the latency sensitivity levels of the jobs. Latency sensitivity levels of the jobs determines the allowed process time within which the workload has to be executed by the provider for the purpose of maintaining the QoS. It has been observed in our earlier work [12] that latency sensitivity levels of the workloads directly affect their energy consumption levels. Higher the latency sensitivity levels of the workloads, more is the energy consumption and less is the allowed process time. From the 17 jobs submitted by User 1, 10 jobs are assigned with a prediction weight of level 3, 4 jobs with level 1, and 3 jobs with level 0 respectively by the rule miner.

Table 4. Predictability Weight for Jobs submitted by User 1

Job Name	Number of Submissions	Event Proportion (%)	Job Predictability Weight ($\boldsymbol{P_{sl}}$)	Latency Level
Job 1	102	28.25	Level 3	2
Job 2	102	28.25	Level 3	0
Job 3	102	28.25	Level 3	2
Job 4	11	3.04	Level 3	2
Job 5	11	3.04	Level 3	0
Job 6	11	3.04	Level 3	2
Job 7	3	0.83	Level 1	1
Job 8	3	0.83	Level 1	0
Job 9	3	0.83	Level 1	2
Job 10	3	0.83	Level 1	2
Job 11	2	0.55	Level 3	0
Job 12	2	0.55	Level 3	0
Job 13	2	0.55	Level 3	0
Job 14	1	0.277	Level 0	0
Job 15	1	0.277	Level 0	0
Job 16	1	0.277	Level 0	0
Job 17	1	0.277	Level 3	1

Now, the following sections of model validation demonstrates the integral process of InOt-RePCoN aimed at predicting user behavioural trend of User 1 whilst submitting Job 1.

6.4 Outlier Detection

From the statistical analysis conducted on the submission of Job 1 from User 1, a total of 105, 80 and 102 submissions of Job 1 are observed in $W_{c,}$ $W_{1,}$ W_2 respectively. The validator computes the presence of outliers in the current window sample before performing the similarity analysis. Fig. 4 presents the presence of outliers contained in the consecutive submission interval trend of Job 1 submitted by User 1 in W_c detected based on (5). The rectangular box depicts the normal distribution of the data, with the solid line in the rectangular box representing the median, and the circles illustrate the outliers. Far the presence of an outlier from the median, more is the deviation of that corresponding outlier from the actual observation. It is evident from Fig. 4 that the submission interval trend of Job 1 from User 1 suffers from significant proportions of outliers in W_c, insisting the need for robust regression process for the purpose of restraining the effects of the influential outliers.

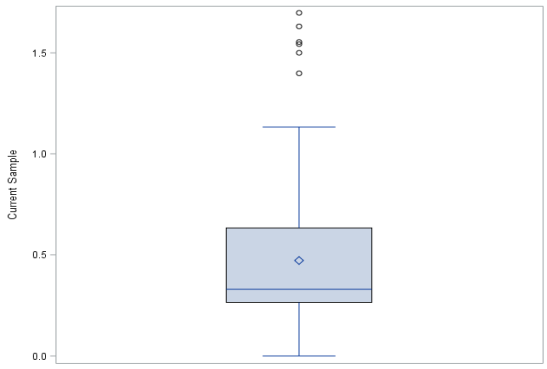
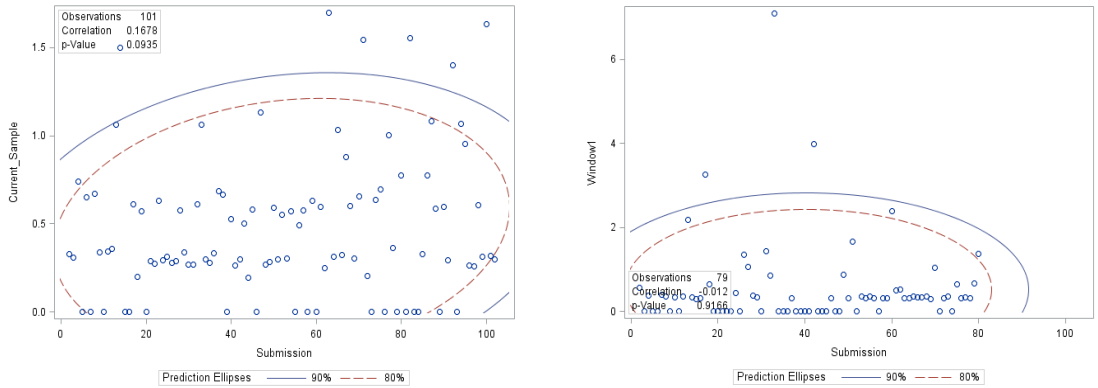


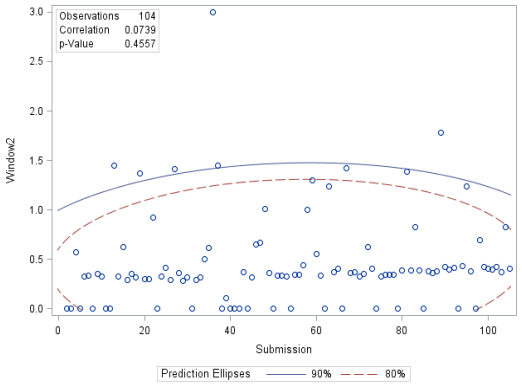
Fig. 4. Presence of outliers for Job 1 of User 1 in Wc

6.5 Estimation of Ellipses



(a)

(b)



(c)

Fig. 5. Prediction ellipses of Job 1 of User 1 (a) Wc (b) W1 (c) W2

Now, the sample validator computes the confidence and prediction ellipses for the submission interval trend of Job 1 of User 1 in W_c, W_1 and W_2 respectively. The value of α in (7) and (8) are set as 0.10 and 0.20 respectively to generate the prediction ellipses of 90% and 80% confidences respectively. The effect of the presence of outliers in the window samples is directly proportional to the number of residuals falling beyond the prediction ellipses. Thus the outliers falling beyond the prediction ellipses are the influential outliers which significantly affect the prediction accuracy. Fig. 5 presents the prediction ellipses generated for the job submission trend of Job 1 submitted by User 1 in all the three Windows.

Table 5. Statistics of Prediction Ellipses

Sample	Number of Submissions	Minimum Interval (μs)	Maximum Interval (μs)	Pearson Coefficient	Mean	Standard deviation

Current	105	1939	101880483	0.16779	28347893	24153527
Window 1	80	1970	425730805	-0.01197	32610565	61543070
Window 2	102	2614	179952098	0.07393	27260821	27941765

The statistics of Pearson correlation coefficient analysis for job submission interval for Job 1 of User 1 are presented in Table 5, along with the analysis results of the prediction ellipses. From Fig. 5 and Table 5, it is evident that the submission interval of Job 1 of User 1 in W_c and W_2 is exhibiting a positive correlation and W_1 is exhibiting a negative correlation. From the prediction ellipses, a close correlation is evident between W_c and W_2 in terms of the prediction confidences and the presence of residuals, which is further validated by the Pearson correlation coefficient. Thus it can be concluded that the trend of Job 1 of User 1 in W_2 is exhibiting a more correlated behaviour with those in W_c than that of W_1, insisting that Job 1 of User 1 predominantly satisfies time-of-the-day periodicity effect. Based on this preliminary analysis for periodicity, the predictor relies on W_2 (validated by the sample validator) for the purpose of further training the most suitable historical sample into the predictor.

6.6 Stationarity Test

Fig. 6 presents the job submission trend, auto-correlation (ACF) and partial-autocorrelation functions (PACF) estimated by the ADF test for the original series of job submission time. Fig. 6 shows a gradual decaying ACF function and also a strong first lag in the PACF with no other significant lags. Table 6 presents the ADF statistics of the stationarity test for Job 1 of User 1. Tau is the test statistics of the ADF unit root test with a standard mean in the data points. The ADF test statistics leads us to infer that the job submission trend is non-stationary since the P value is greater than 0.05, the null-hypothesis cannot be rejected and so the trend of job submission trend is non-stationary.

Table 6. ADF test for Job submission time

Type	Value
Tau (Single Mean)	1.56
Tau (Trend)	-1.58
P value (Single mean)	0.9994
P value (Trend)	0.7929

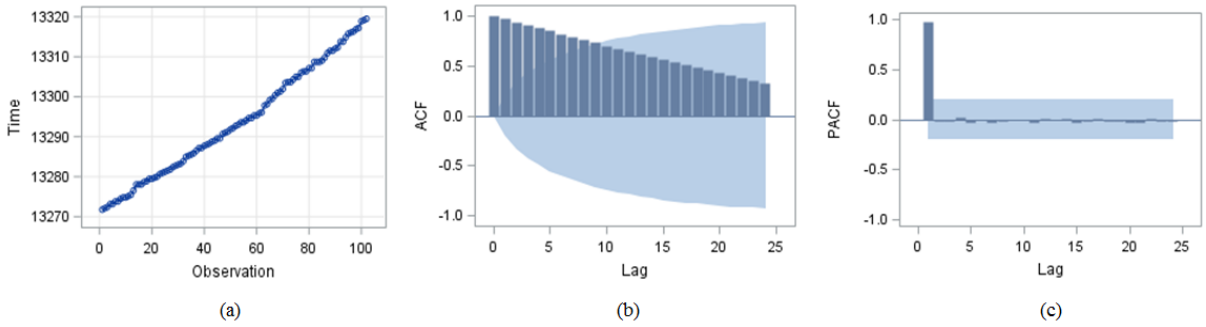
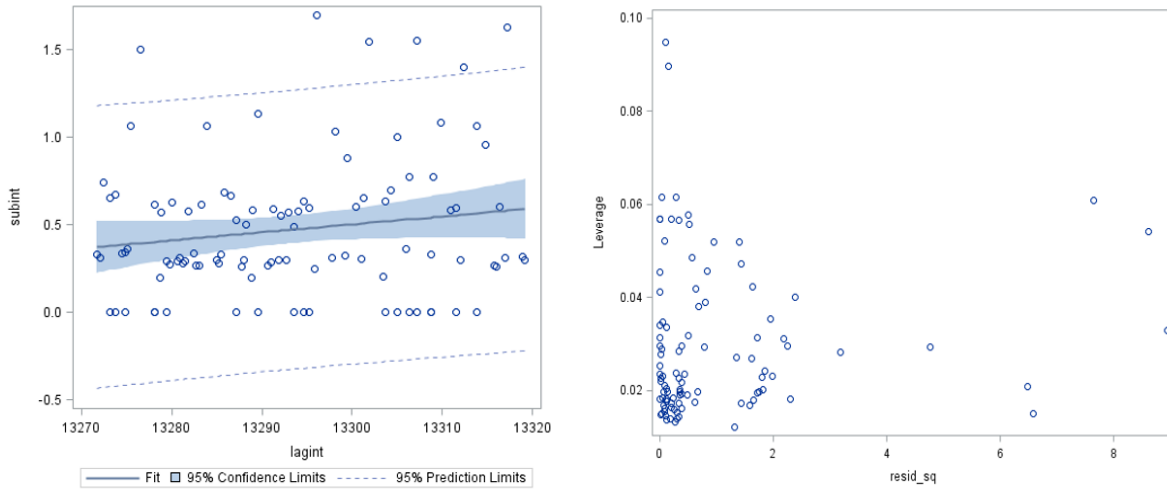


Fig. 6. ADF test for job submission time (a) Submission trend (b) ACF (c) PACF

6.7 Outlier Suppression

Fig. 7 illustrates the regression fit plot for the job submission interval of Job 1 of User 1 in W_c, fitted with a 95% confidence and prediction limits for the submission interval trend along with the leverage-to-residual square plot. It can be easily observed that the submission interval trend of Job 1 of User 1 is heavily influenced by the presence of a significant number of outliers, which could lead to inaccurate prediction of the job submission trend. The submission interval trend is suffering from both high leverages and large residuals, necessitating the need for suppressing the influence of the outliers with robust regression.



(a) (b)

Fig. 7. Regression for Job 1 of User 1 (a) Fit plot (b) Leverage to residual square plot

Table 7 further presents the observations of Job 1 of User 1 suffering significantly from the presence of outliers in the submission interval trend in W_c identified by the robust regression. The level of influence of the outliers over the data points are determined by the Cook's distance which is a combined measure of leverage and residuals present in the observation. Apart from the observations presented in Table 7, presence of all the outliers have a considerable impact on the overall prediction accuracy depending on their distance from the mean. Thus the predictor considers suppressing the influence of all the outliers by adjusting their weights depending on their corresponding influence on prediction accuracy.

Table 7. Influential Outliers in the submission interval trend of Job 1 of User 1

Submission	Time	Submission Interval	Cook Distance > 4/102	Weight
14	13278.06	1.50030	0.10112	0.33012
63	13297.82	1.69801	0.16488	0.35201
82	13308.70	1.55352	0.04620	0.36555
92	13313.76	1.39880	0.04795	0.41859
100	13318.86	1.63076	0.16428	0.31991

Now the predictor subjects the prediction sample of Job 1 of User 1 in W_c to robust regression as described in section 5.3.2. Robust regression is now applied on the prediction sample by iterated re-weighted least squares based on the Huber weights. Observations highly suffering from the outliers are assigned with smaller weights by robust regression in order to suppress their influence on the prediction accuracy. After applying robust regression to supress the outliers in the prediction sample of Job 1 of User 1 in W_c, the outlier proportions has been reduced to 0.0594%. Table 7 further presents the weights assigned to the observations dominated heavily the influential outliers. It can be observed that observations influenced by severe outliers are assigned with lower weights through to a weight of one is assigned to the observations under minimum outlier influence. Finally, the prediction sample is sorted ascendingly based on the weights assigned to the observations by robust regression for the purpose of restraining the influence of the outliers contained in the prediction sample.

6.8 ARIMA Model Selection

Since the original variable of the job submission time for Job 1 of User 1 is non-stationary, the predictor now computes the differenced variable of submission interval for Job 1 of User 1. Table 9 presents the test statistics of stationarity for the differenced variable after subjecting to robust regression. Fig. 8 presents the trend, ACF and PACF statistics of the differenced variable for the job submission interval after robust regression. From Table 8, the Tau value for the differenced variable is significantly negative with a very small p value, so that the null hypothesis is rejected and there is alternative hypothesis. It can also be observed that there is no gradual lag in the ACF function and the PACF function is exhibiting a first positive significant lag followed with a second significantly negative lag and no other lags are significant. All these statistics conclude that the differenced

variable which is the submission interval of the Job 1 of User 1 is now stationary. Furthermore, the first positive lag of ACF insist that there is an AR 1 process existing in this stationarised series. It is clear that the differenced variable is more suitable for predicting the future trend because of the degree of stationarity.

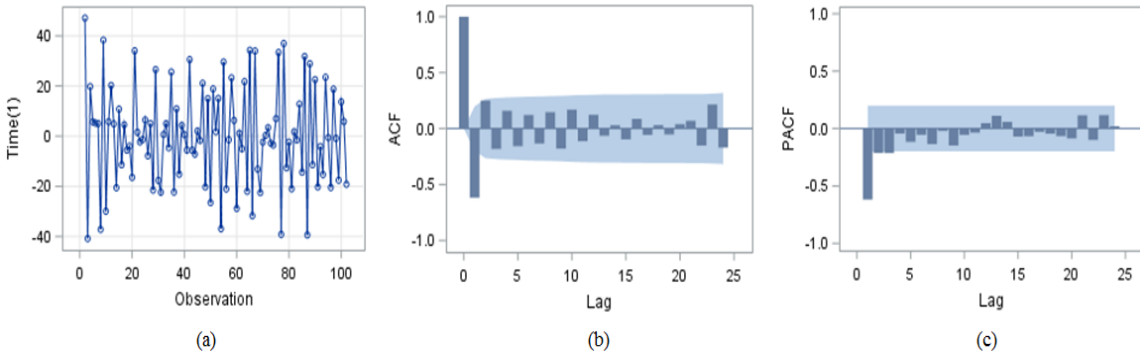


Fig. 8. ADF test for differenced variable (a) Trend (b) ACF (c) PACF

Table 8. ADF test for the differenced variable

Type	Value
Tau (Single Mean)	-21.45
Tau (Trend)	-21.33
P value (Single mean)	<.0001
P value (Trend)	<.0001

Now, the predictor estimates different ARIMA models for the purpose of training the predictor with the most appropriate predictor variables based on the trend of the stationarised differenced variable. Table 9 presents the estimates for various ARIMA models for the job submission interval of Job 1 of User 1. From Table 9, it can be concluded that an ARIMA(1,1,1) model is best suitable for predicting the trend of Job 1 of User 1, since both the AIC and SBC values are smaller than those of the other two models. Further the lags in the residual correlations of both ACF and PACF for ARIMA (1,1,1) are non-significant with only a first significant positive lag in the ACF plot for confidence, as shown in Fig. 9.

TABLE 9. ARIMA model estimates for job submission interval

ARIMA (p,q,d)	Conditional Least Square Estimate			
	Parameter	Estimate	AIC	SBC
ARIMA (1, 1, 0)	AR 1,1	0.57905	65.37405	70.60429
ARIMA (0, 1, 1)	MA 1,1	-0.36224	84.38915	89.61939
ARIMA (1, 1, 1)	AR 1,1	0.65974	31.35396	39.19932
	MA 1,1	1.00000		

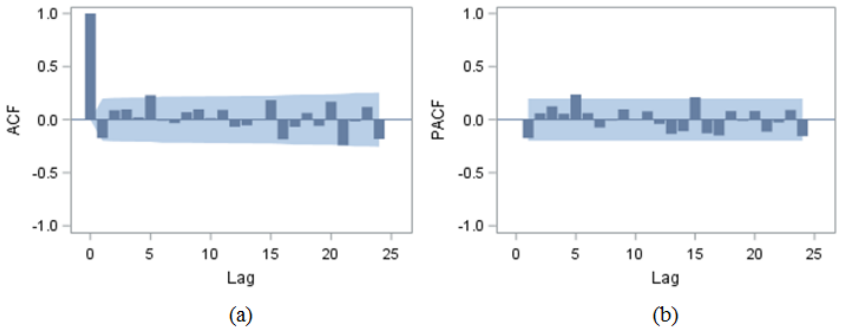


Fig. 9. Predictor plots of ARIMA (1,1,1) (a) ACF (b) PACF

6.9 Optimised Job Trend Prediction

The predictor choses the ARIMA (1,1,1) model for training the predictor for the trend of Job 1 of User 1. After choosing the ARIMA model, the anticipated number of submissions and the session duration for Job 1 of User 1

is predicted to set the forecast window of the predictor based on section 5.3.3, explained as follows. Job 1 of User 1 spans across a total of 102, 105 and 21 submissions in $W_{c,}$ $W_{2,}$ and W_r respectively. The session duration of Job 1 of User 1 in $W_{c,}$ $W_{2,}$ and W_r are 47.71, 47.25 and 11.72 respectively. In other words, User 1 has submitted Job 1 for 102 times across a duration of 47.71 minutes in W_c. From W_r, predictor obtains the composite $V_r = \{21, 11.72\}$, where 21 is the number of submissions and 11.72 is the submission session duration for Job 1 of User 1 in W_r. Based on the computations presented in section 6.3.3, the window forecaster computes the anticipated number of submissions and the session duration for Job 1 of User 1 as a composite with the absolute values of $V_p = \{23, 11.72\}$ for the prediction window W_p. Now the predictor predicts the submission interval of Job 1 of User 1 using the chosen ARIMA (1,1,1) model based on V_p. Fig. 10 presents the ARIMA forecast for Job 1 of User 1 for the prediction window W_p. This is a linear forecast with a 95% confidence window for the submission interval of Job 1 of User 1 for the anticipated 23 submissions in W_r. It can be observed that the 95% confidence window spans across a significant interval bounds across the linear forecast.

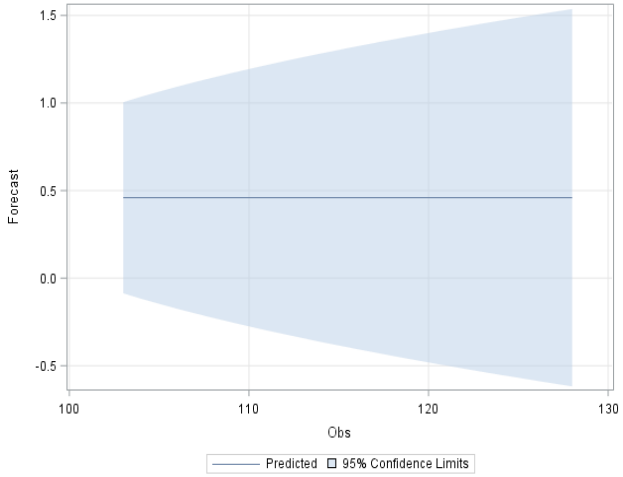


Fig. 10. ARIMA forecast for Job 1 of User 1

InOt-RePCoN further optimises this 95% confidence interval of the ARIMA forecast in order to improve the prediction accuracy and to reduce the interval bounds around the linear forecast. Fig. 11 presents the optimised confidence interval based on section 5.3.5. It can be observed that the ARIMA forecast is further optimised after nullifying the negative lower bounds with an optimised upper confidence interval. The upper confidence interval W_{con} is the optimised forecast with an error margin of the forecast expected with the bounds of W_{con} and zero mean lower bound, owing to the presence of outliers in the prediction sample. This insists that the future submission interval of Job 1 of User 1 is expected within the bounds of W_{con} and the zero mean lower bound limits.

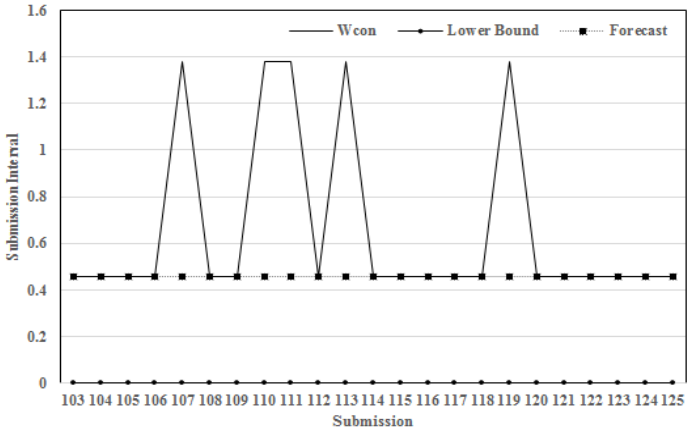


Fig. 11. Optimised confidence window

7. Performance Evaluation

The efficiency of our proposed prediction model is evaluated by the measure of the forecast accuracy against the actual trend of the user behaviours in terms of the anticipated number of submissions, session duration and the submission interval for the target jobs and users. The efficiencies of our proposed model is evaluated under various scenarios of business hours in order to demonstrate the dependency of InOt-RePCoN under dynamic scenarios of Cloud Computing.

7.1 Week Day Off-Peak Time Prediction

7.1.1 Sample containing influential outliers

The sample trained in section 6 contains data from Wednesday during the business hours of 12 am to 1 am for the purpose of predicting the expected user behaviour from 1 am to 2 am, which is an off-peak business hour during a week day. Fig. 12 illustrates the accuracy of our forecasting window, where the actual observations of the number of job submissions and the session duration are plotted against the forecasted values for Job 1 of User 1. It can be observed that an accuracy of 88.46% is achieved for the number of job submissions and an accuracy of 91.13% is achieved whilst predicting the session duration for job 1 of User 1.

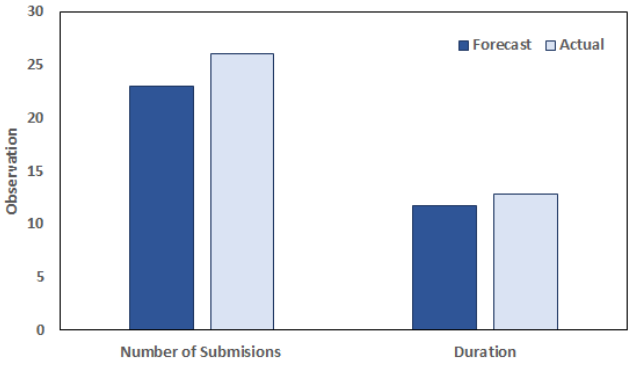


Fig. 12. Forecast window observation

Fig. 13 presents the optimised confidence interval of the InOt-RePCoN fitted with the actual trend of the submission interval for Job 1 of User1. Since the prediction sample is heavily affected by the presence of influential outliers, the future trend is expected to be within the bounds of W_{con} and the zero mean lower bound error margin. It can be observed that 18 observation points out of 23 of the actual submissions are within the confidence interval bound limits predicted by our proposed model. Thus our proposed model achieves an accuracy of 78.26% whilst predicting the submission interval trend of Job 1 of User 1. The confidence bounds optimised by our proposed model significantly reduces the 95% interval of the ARIMA forecast, thus reducing the bound limits with reliable level of accuracy.

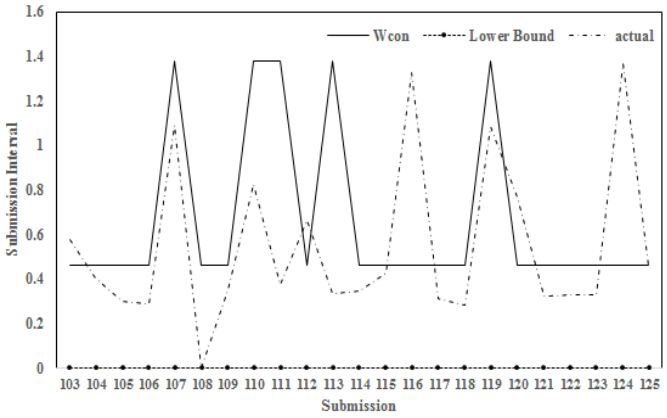


Fig. 13. Optimised confidence

7.1.2 Samples without Influential outliers

The efficiency of our proposed model are further evaluated whilst forecasting the user behavioural trend with the sample containing no influential outliers during off-peak time. Now the prediction model is trained with the data obtained from Wednesday during the business hours of 3 am to 4 am for the purpose of predicting the expected user behaviour from 4 am to 5 am. The rule miner forms W_1 and W_2 accordingly as explained in section 5.1. A total of 57 users are observed in W_c, with 39, 5, 4, and 9 users are assigned with a predictability weight of level 3, 2, 1 and 0 respectively by the rule miner. A randomly chosen user (named User 2) has been set as the target user for this forecast. User 2 has submitted a total of 3 job types across 36 submissions. Both User 2 and his three job types are assigned with a predictability weight of level 3. Now the objective is set to the predictor to forecast the trend of all the three job types belonging to User 2. Robust regression is not performed by the predictor in spite of the minimal presence and marginal influence of the outliers in the job submission trend of the prediction sample.

The parameter for the forecast window is computed as an absolute composite of V_p = {36, 56.89}, which means a total of 36 submissions are anticipated from User 2 in 56.89 minutes in W_p. Fig. 14 shows the ARIMA forecast for the submission interval trend of User 2 bounded with the 95% confidence interval, along with the forecast fitted with the actual observation. It can be observed that the occurrence of peaks and valleys of the forecast is closely correlating with the actual trend, but the forecast-to-actual values are still not accurately optimised. Furthermore, the 95% confidence of the ARIMA forecast is spanning across the forecast with a larger amplitude, which reduces the crispness of the forecast results.

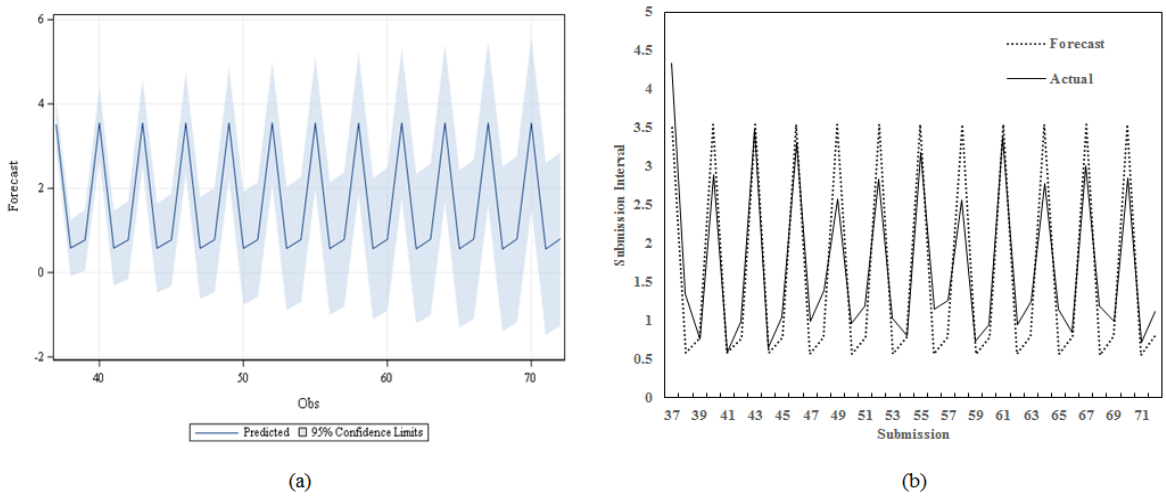


Fig. 14. Forecast for all jobs of User 2 (a) ARIMA forecast (b) Forecast vs Actual trend

Fig. 15 illustrates the number of submissions and the session duration predicted by our forecasting window against the actual values observed. It can be observed that an accuracy of 100% is achieved in forecasting the anticipated

number of submissions and an accuracy of 96.71% is achieved in forecasting the session duration for the trend of User 2.

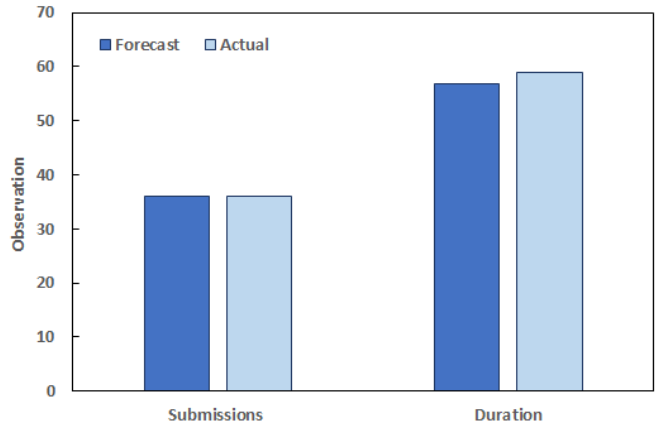


Fig. 15. Forecast window observation

Fig. 16 depicts the optimised confidence interval W_{con}, fitted with the actual trend of submission interval of User 2. It can be observed that most of the actual trend of the submission interval of User 2 closely correlated with the optimised confidence interval, with W_{con} achieving an accuracy of 73.23% delivered by our proposed prediction model. It is also evident that the optimised confidence interval of our proposed model significantly enhances the prediction accuracy of the initial ARIMA forecast.

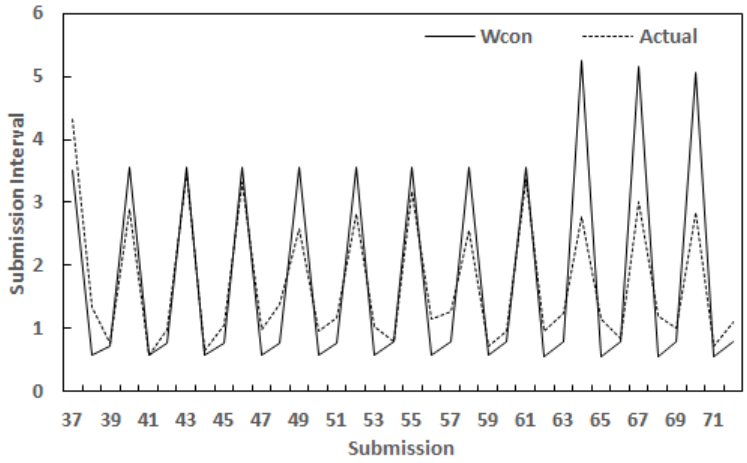


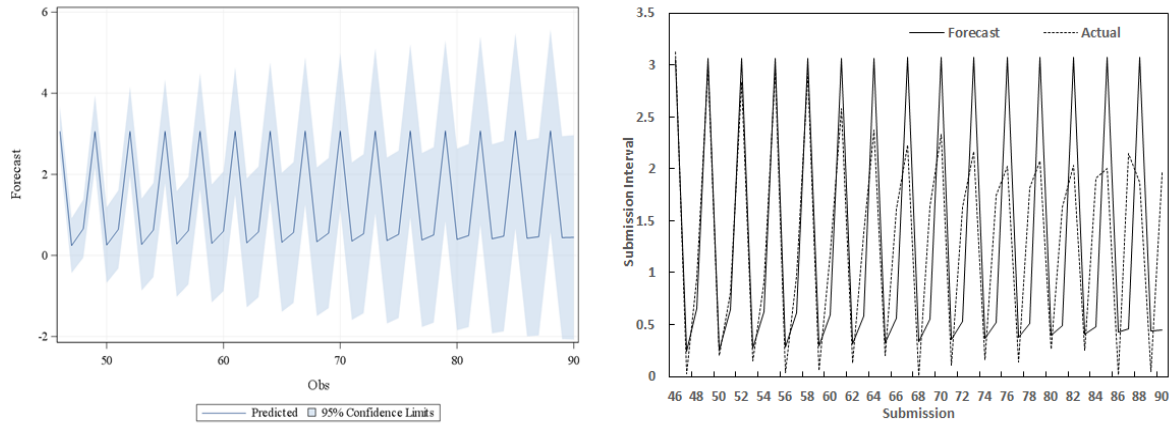
Fig. 16. Optimised confidence

7.2 Week Day Peak Time Prediction

Further, the efficiency InOt-RePCoN is evaluated whilst predicting the expected user behaviours under peak business hours during a week Day. Now, the objective is to forecast the user behaviour from 11 am to 12 pm on a Monday morning. A total of 58 users are comprised in W_c, with 43, 4, 7, and 4 users are assigned with a predictability weight of 3, 2, 1, and 0 respectively. This user (named User 3), has submitted the most number of jobs in W_c, has been set as the target user for this forecast. User 3 has submitted a total of 2 job types named Job 1 and Job 2 respectively, across a total of 75 job submissions in W_c. Both these two job types has been assigned with a predictability weight of level 3 by the rule miner, insisting an increased predictability. The event proportion for Job 1 and Job 2 is 60 and 40% respectively. Now predictor is set with an objective of forecasting the future trend of Job 1 of User 2.

A significant variance is evident in the submission interval of Job 1 of User 3 with a minimal influence of outliers in W_c. After generating the prediction ellipses, a close correlation is evident in the submission trend of Job 1 of

User 3 between W_c and W_1. Thus the dual-effect window shows better correlation and exhibits a better trend of predictability for Job 1 of User 3. This is because the dual effect window contains the historic sample from the same representative Monday from the previous week. But W_2, the time-of-the-day effect window consists of sample from a Sunday (previous day of the current sample). Since W_2 contains week-end trend it is loosely correlating with the trend in W_c containing week-day trend. Thus the samples in W_1 is chosen and validated for similarity by the validator for the purpose of further training into the predictor. The parameter for the forecast window is computed as an absolute composite of V_p = {45, 58.09}, which means a total of 45 submissions are anticipated for Job1 of User 3 in 58.09 minutes in W_p. Fig. 17 presents the ARIMA forecast output for the submission interval trend of Job 1 of User 3 with the 95% confidence interval, alongside the forecast fitted with the actual observation of submission interval. Again, the occurrence of peaks and valleys of the forecast is closely correlating with the actual trend, but the forecast-to-actual values are still not accurately optimised.



(a) (b)

Fig. 17. Forecast for all Job 1 of User 3 (a) ARIMA forecast (b) Forecast vs Actual trend

Fig. 18 illustrates the number of submissions and the session duration predicted by our forecasting window against the actual values. It can be observed that an accuracy of 100% is achieved in forecasting the anticipated number of submissions and an accuracy of 96.76% is achieved in forecasting the session duration for the trend of Job 1 of User 3.

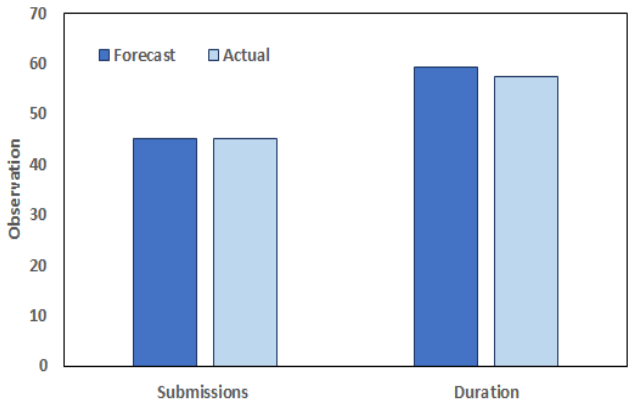


Fig. 18. Forecast window observation

Fig. 19 depicts the optimised confidence interval fitted with the actual trend of submission interval of Job 1 of User 3. It can be observed that most of the actual trend of the submission interval of Job 1 of User 3 are within the bounds of the optimised confidence interval, with W_{con} achieving an accuracy of 82.22% delivered InOt-RePCoN.

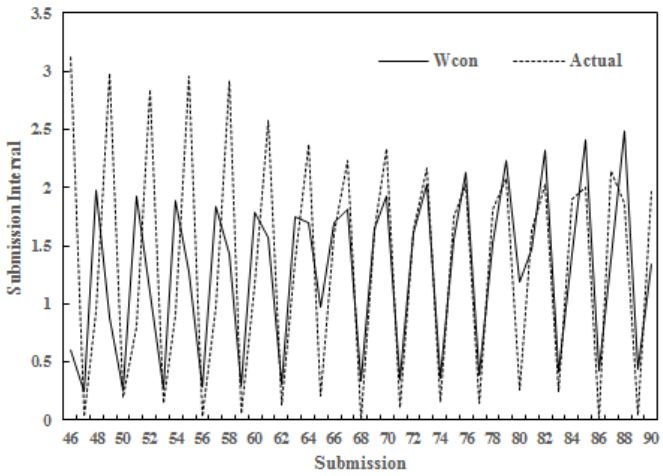


Fig. 19. Optimised confidence

7.3 Reduction of Under-Prediction

Form the perspectives of benefitting both the users and the providers, InOt-RePCoN is aimed at reducing the probabilities of under-prediction, since it would cause a more disastrous effect on the overall energy efficiency. Fig. 20 presents the over-to-under predicted ratio whilst forecasting the submission interval of jobs from users during peak and off-peak business hours. It can be observed that our proposed prediction model is effective in reducing the number of under-predictions, witnessed only at an average of 27.45% in comparison to the over-predicted observations witnessed at an average of 74.01%. Thus it can be concluded that our proposed prediction model is effective in reducing the probabilities of under-prediction.

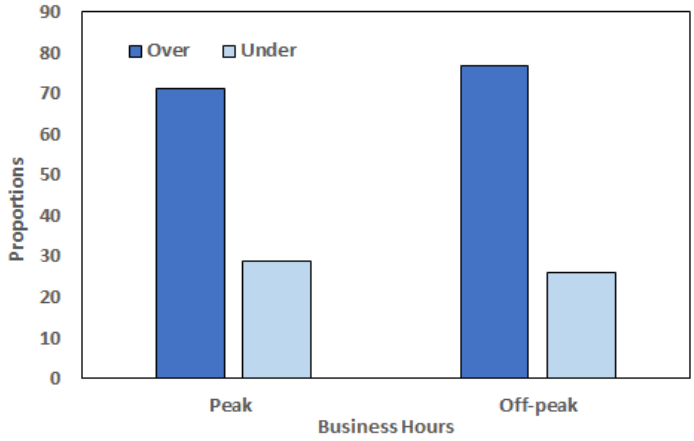


Fig. 20. Over-to-under prediction ratio of submission interval

7.4 Forecasting Efficiency of InOt-RePCoN

This section is aimed at demonstrating the forecast efficiency of InOt-RePCoN, by evaluating the forecast accuracy of our proposed model against existing benchmarks. Since the objectives of InOt-RePCoN are to predict the intensity of the incoming job submissions rather than estimating the resource consumption levels of the arrived workloads, the benchmark techniques are chosen such that they have similar prediction objectives of forecasting the trend of job arrival.

Firstly the accuracy of the statistical approach adopted by InOt-RePCoN has been evaluated against SPAR (Spare Periodic Auto Auto-Regression) which is an autoregressive based prediction model, an approach of predicting

load on single VM based on HMM using single time series and a HMM based co-clustering technique of predicting workloads at group levels using multiple time series respectively, since all these techniques are aimed at predicting the job arrival trend. This evaluation is intended to demonstrate the efficiency of our proposed Influential Outlier Restrained Prediction based on ARIMA integrated with our novel Confidence Optimisation framework. Fig. 21 illustrates the prediction accuracy of InOt-RePCoN against the chosen prediction models. It can be observed that our proposed model outperforms the other three statistical approaches exhibiting an average prediction accuracy of 76.73%, against the multiple time series co-clustering HMM, Spare Periodic Auto-Regression, and single time series HMM technique at 73%, 68% and 55% respectively. By delivering a reliable level of accuracy for the confidence window, service providers can expect the consecutive submission of the corresponding jobs from the users within the window intervals predicted by InOt-RePCoN.

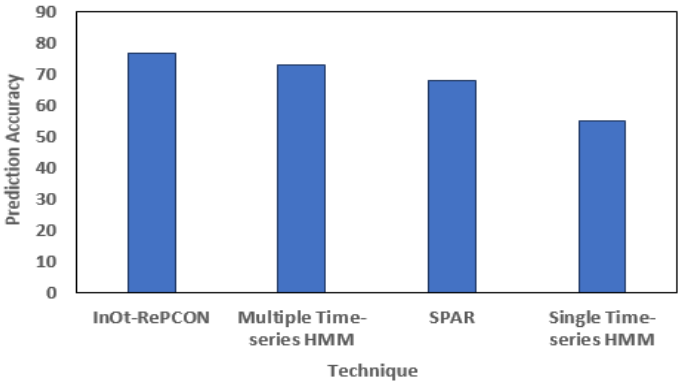


Fig. 21. Confidence optimisation accuracy

Secondly, the prediction accuracy of InOt-RePCoN is evaluated whilst forecasting the anticipated number of submissions and the session duration for the users by the way of evaluating the average prediction error of our proposed prediction model against the existing RPPS based on simple ARIMA and SPAR based on periodic Auto-Regression, both aimed at forecasting the incoming job tend based on different adoptions of Auto-Regression. The error percentage of our proposed prediction model is presented as a combination of the average prediction error whilst forecasting the anticipated number of submissions and the session duration. The prediction error of the other two models are presented as a combination of the average under and over prediction errors whilst forecasting future workload levels. Fig. 22 depicts the average prediction error of InOt-RePCoN, RPPS and the SPAR model respectively, it is evident that our proposed prediction model exhibits a better prediction accuracy with an average prediction error of 11.79, than the RPPS and SPAR models with an average prediction error of 14.39 and 17.68 respectively. With both the number of submissions and the session duration being predicted with a reliable level of accuracy, service providers can achieve an effective scaling of the server resources based on the anticipated intensity of the incoming job trend.

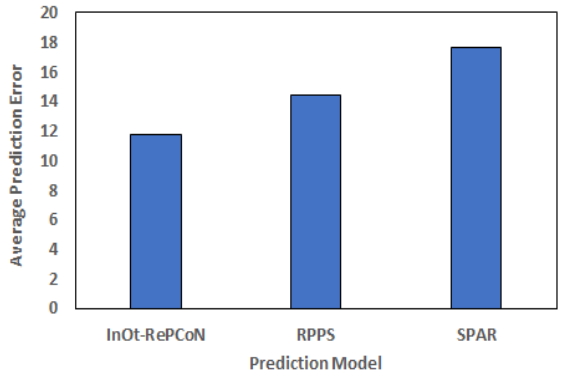


Fig. 22. Prediction efficiency

8. Conclusion

This paper proposes InOt-RePCoN, a novel prediction model for forecasting the trend of user behaviours in large-scale Cloud environments. Our proposed model is expected to benefit the service providers in two different perspectives. Firstly, predicting the expected number of submissions and session duration for users helps the service providers to achieve an optimum resource management by scaling up/down the server resources in accordance with the window forecast. For instance, accurately predicting the peaks and valleys of the workload levels from the users helps in effective switching of the server resources for energy efficient server management. Secondly, the optimised confidence interval forecast for the submission interval trend of the users provides useful inferences about the arrival frequency of the jobs from the users. This helps with an initial preparation for the scheduling and job allocation management in accordance with the arrival rate of the workloads from the users. Our proposed model exhibits a characteristic reduction in the probabilities of under-predictions which helps avoiding the energy expenditures incurred by the early provisioning of resources for the anticipated job submissions. Thus our tri-fold forecast helps to accurately model the user behavioural trend in Cloud environments. From the performance evaluations conducted based on real Cloud traces during different business hours, we conclude that our proposed prediction model achieves reliable level of accuracy in predicting the future job submission trend of the users. Our proposed model outperforms the existing prediction models based on simple auto-regression, simple ARIMA and co-clustering time-series techniques in terms of the reduced average prediction error and increased prediction accuracy. One notable complexity of our model could be attributed to the need for storing the historical samples in the database, since our model utilises the historical samples to optimise the confidence interval of the ARIMA forecast. But the complexities in storing the historical samples is reduced to a minimum, since our model needs historical samples from just two previous weeks. Since our proposed model relies on the inherent periodicity among the users and their workloads for computing their predictability weights and further optimising the prediction confidence, the prediction of user behaviours without any degree of periodicity, essentially brand new users, cannot be further optimised by our model. Though, recoding the traces of such users over a period may facilitate optimising their prediction confidence.

The effects of the influential outliers in the prediction sample have been restrained by our proposed model to reduce the impacts of such outliers over the forecast accuracy. Still, the submission interval forecast of our proposed model for the samples containing no outliers is better than the forecast for samples containing significant proportions of outliers during both off-peak and peak times. For instance, the occurrence of peaks and valleys in the future submission intervals of prediction samples containing no outliers are forecasted more accurately than the samples influenced by the presence of outliers. In other words, the error margin for samples containing no outliers are much lower than the samples influenced by outliers. As a future work, we will investigate the possibility of enhancing the prediction accuracy of InOt-RePCoN, with the motivation of reducing the prediction error margin interval for outlier influenced samples. Furthermore, it is increasingly common in a datacentre environment that the allocated resource levels for workload execution far exceed the minimum requirement to complete the task. Over-allocating the resource levels are vulnerable to leave most of the allocated server resources idle without actually contributing towards workload execution causing undesirable energy consumptions. To this end, we plan to develop a prediction framework to forecast the resource requirements of the incoming workloads for the purpose of benefiting optimum resource provisioning in the datacentres to promote sustainable datacentre execution.

Acknowledgement

This work was partially supported by the National Natural Science Foundation of China under Grants No. 61502209 and 61502207.

References

[1] S.-Y. Jing, S. Ali, K. She, and Y. Zhong, "State-of-the-art research study for green cloud computing," *The Journal of Supercomputing*, vol. 65, pp. 445-468, 2013.

[2] J. Patel, V. Jindal, I.-L. Yen, F. Bastani, J. Xu, and P. Garraghan, "Workload Estimation for Improving Resource Management Decisions in the Cloud," 12th International Symposium on Autonomous Decentralized Systems, Taichung, 2015.

[3] L. Liu, D.-A. DaSilva, N. Antonopoulos, Z. Ding, and Y. Zhan, "Achieving Green IT Using VDI in Cyber Physical Society," *Journal of Internet Technology*, vol. 14, pp. 413-424, 2013.

[4] J. Li, B. Li, T. Wo, C. Hu, J. Huai, L. Liu, *et al.*, "CyberGuarder: A Virtualization Security Assurance Architecture for Green Cloud Computing," *Future Generation Computer Systems*, vol. 28, pp. 379–390, 2012.

[5] J. Panneerselvam, L. Liu, N. Antonopoulos, and B. Yuan, "Workload Analysis for the Scope of User Demand Prediction Model Evaluations in Cloud Environments," 7th International Conference on Utility and Cloud Computing (UCC), London, 2014.
[6] I. S. Moreno and J. Xu, "Customer-Aware Resoruce Overallocation to Improve Energy Efficiency in Real-Time Cloud Computing Data Centres," International Conference on Service-Oriented Computing and Applications (SOCA), Irvine, 2011.
[7] Q. Zhang, M. F. Zhani, S. Zhang, Q. Zhu, R. Boutaba, and J. L. Hellerstein, "Dynamic Energy-Aware Capacity Provisioning for Cloud Computing Environments," Proceedings of the 9th international conference on Autonomic computing, San Jose 2012.
[8] 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, pp. 755-768, 2012.
[9] B. Ciciani, D. Didona, P. D. Sanzo, R. Palmieri, S. Peluso, F. Quaglia, *et al.*, "Automated Worklaod Characterization in Cloud-based Transactional Data Grids," 26th InternationalParallel and Distributed Processing Symposium Workshops & PhD Forum, Shanghai, 2012.
[10] S. Mahambre, P. Kulkarni, U. Bellur, G. Chafle, and D. Deshpande, "Workload Characterization for Capacity Planning and Performance Management in IaaS Cloud," International Conference on Cloud Computing in Emerging Markets (CCEM), Bangalore, 2012.
[11] A. Khan, X. Yan, S. Tao, and N. Anerousis, "Workload characterization and prediction in the cloud: A multiple time series approach," IEEE Network Operations and Management Symposium, Maui, 2012.
[12] J. Panneerselvam, L. Liu, N. Antonopoulos, and M. Trovati, "Latency-Aware Empirical Analysis of the Workloads for Reducing Excess Energy Consumptions at Cloud Datacentres," IEEE Symposium on Service-Oriented System Engineering (SOSE), Oxford, 2016.
[13] I. S. Moreno, P. Garraghan, P. Townend, and J. Xu, "An Approach for Characterizing Workloads in Google Cloud to Derive Realistic Resource Utilization Models," 7th International Symposium on Service Oriented System Engineering (SOSE), Redwood City, 2013.
[14] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, *et al.*, "Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services," Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation, 2008.
[15] W. Fang, Z. Lu, J. Wu, and Z. Cao, "RPPS: A Novel Resource Prediction and Provisioning Scheme in Cloud Data Center," IEEE 9th International Conference on Services Computing, Honolulu, HI, 2012.
[16] J. Yang, C. Liu, Y. Shang, Z. Mao, and J. Chen, "Workload Predicting-Based Automatic Scaling in Service Clouds," 6th International Conference on Cloud Computing, Santa Clara, CA, 2013.
[17] S. Mallick, G. Hains, and C. S. Deme, "A Resoruce Prediction Model for Virtualization Servers," International Conference on High Performance Computing and Simulation (HPCS), Madrid, 2012.
[18] T.-T. Duy, Y. Sato, and Y. Inoguchi, "Performance evaluation of a green scheduling algorithm for energy savings in cloud computing," in *Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium on*, 2010, pp. 1-8.
[19] A. A. Bankole and S. A. Ajila, "Cloud Client Prediction Models for Cloud Resource Provisioning in a Multitier Web Application Environment," 7th International Symposium on Service-Oriented System Engineering, Redwood City, 2013.
[20] 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, pp. 155-162, Januray 2012.
[21] I. S. Moreno, P. Garraghan, P. Townend, and J. Xu, "Analysis Modleling and Simulation of Workload Patterns in a Large Scale Utility Cloud," *IEEE Transactions on Cloud Computing*, vol. 2, pp. 208 - 221, 02 April 2014.
[22] E. Caron, F. Desprez, and A. Muresan, "Forecasting for Grid and Cloud Computing On-Demand Resources Based on Pattern Matching," 2nd IEEE International Conference on Cloud Computing Technology and Science, 2010.
[23] T. Li, J. Wang, W. Li, T. Xu, and Q. Qi, "Load Prediction-based Automatic Scaling Cloud Computing," International Conference on Networking and Network Applications, 2016.
[24] Y. Hu, B. Deng, F. Peng, and D. Wang, "Workload Prediction for Cloud Computing Elasticity Mechanism," IEEE International Conference on Cloud Computing and Big Data Analysis, 2016.
[25] C.-F. Wang, H. Wen-Yi, and C.-S. Yang, "A Prediction Based Energy Conserving Resources Allocation Scheme for Cloud Computing," IEEE International Conference on Granular Computing (GrC), Noboribetsu, 2014.
[26] S. Pacheco-Sanchez, G. Casale, B. Scotney, S. McClean, G. Parr, and S. Dawson, "MarkovianWorkload Characterization for QoS Prediction in the Cloud," International Conference on Cloud Computing (CLOUD), Washington, 2011.
[27] S. Di, D. Kondo, and W. Cirne, "Host Load Prediction in a Google Compute Cloud with a Bayesian Model," International Conference for High Performance Computing, Networking, Storage and Analysis (SC), Salt Lake City, 2012
[28] S. Di, D. Kondo, and W. Cirne, "Google hostload prediction based on Bayesian model with optimized feature combination," *Journal of Parallel and Distributed Computing*, vol. 74, pp. 1820-1832, January 2014.
[29] W. Iqbal, M. N. Dailey, D. Carrera, and P. Janecek, "Adaptive resource provisioning for read intensive multitier applications in the cloud", Future Generation Computer System," *Future Generation Computer Systems*, vol. 27, pp. 871-879, June 2011.
[30] N. K. Gondhi and P. Kailu, "Prediction Based Energy Efficient Virtual Machine Consolidation in Cloud Computing," 2nd International Conference on Advances in Computing and Communication Engineering, 2015.
[31] Y. Shen, "Virtual resource scheduling prediction based on a support vector machine in cloud computing," 8th International Symposium on Computational Intelligence and Design, 2015.

[32] N. Roy, A. Dubey, and A. Gokhale, "Efficient Autoscaling in the Cloud using Predictive Models forWorkload Forecasting," IEEE 4th International Conference on Cloud Computing, Washington, 2011.
[33] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch, "Towards Understanding Heterogeneous Clouds at Scale: Google Trace Analysis," Intel Science and Technology Center for Cloud Computing, Pittsburgh, 2012.
[34] Y. Lu, J. Panneerselvam, L. Liu, and Y. Wu, "RVLBPNN: A Workload Forecasting Model for Smart Cloud Computing," *Scientific Programming*, vol. 2016, pp. 1-9, 19 September 2016.
[35] Google. (2011). *Google Cluster Data V2*. Available: https://github.com/google/clusterdata/blob/master/ClusterData2011_2.md
[36] M. Alam, K. A. Shakil, and S. Sethi, "Analysis and Clustering of Worklaods in Google Cluster Trace based on Resource Usage," Cornell University, January 2015.
[37] Z. Liu and S. Cho, "Characterizing Machines and Workloads on a Google Cluster," 41st International Conference on Parallel Processing Workshops, Pittsburgh, PA, 2012.
[38] Y. Jiang, C.-S. Perng, T. Li, and R. Chang, "ASAP: A Self-Adaptive Prediction System for Instant Cloud Resource Demand Provisioning," 11th IEEE International Conference on Data Mining, Vancouver, 2011.
[39] P. Garraghan, P. Townend, and J. Xu, "An Analysis of the Server Characteristics and Resoruce Utilization in Google Cloud," International Conference on Cloud Engineering, Redwood City, 2013.
[40] T. Wang, J. Wei, W. Zhang, H. Zhong, and T. Huang, "Workload-aware anomaly detection for web applications," *The Journal of Systems and Software*, vol. 89, pp. 19-32, March 2014.