Machine Learning with Cloud and Distributed Systems

Investigating the Application of Machine Learning Models to Predict Resource Usage Trends within Distributed Systems

Raphael Leong

Background Overview

Cloud and Distributed Systems

Example of Google Cluster within their data center

Google Cluster Trace

- Dataset that is collected over a period of 29 days
- Outlines the details of around 12,500 machines
- Timeseries data describes cluster usage and job scheduling
- Dataset used in prior works to research cloud and distributed systems

Motivations

- To further understand the complexities behind cluster scheduling
- To improve workload distribution and optimisation between nodes
- Investigate and demonstrate the potential of machine learning models within this space

Objectives

- Analyse the Google cluster trace dataset
- Investigate the relationship between cluster machines based on defined characteristics set by prior works
- Explore the possibility of utilising machine learning models to predict future cluster machine workloads

Cluster Trace Analysis

Initial Observations

Based the analysis on prior observations and research of cluster usage trends:

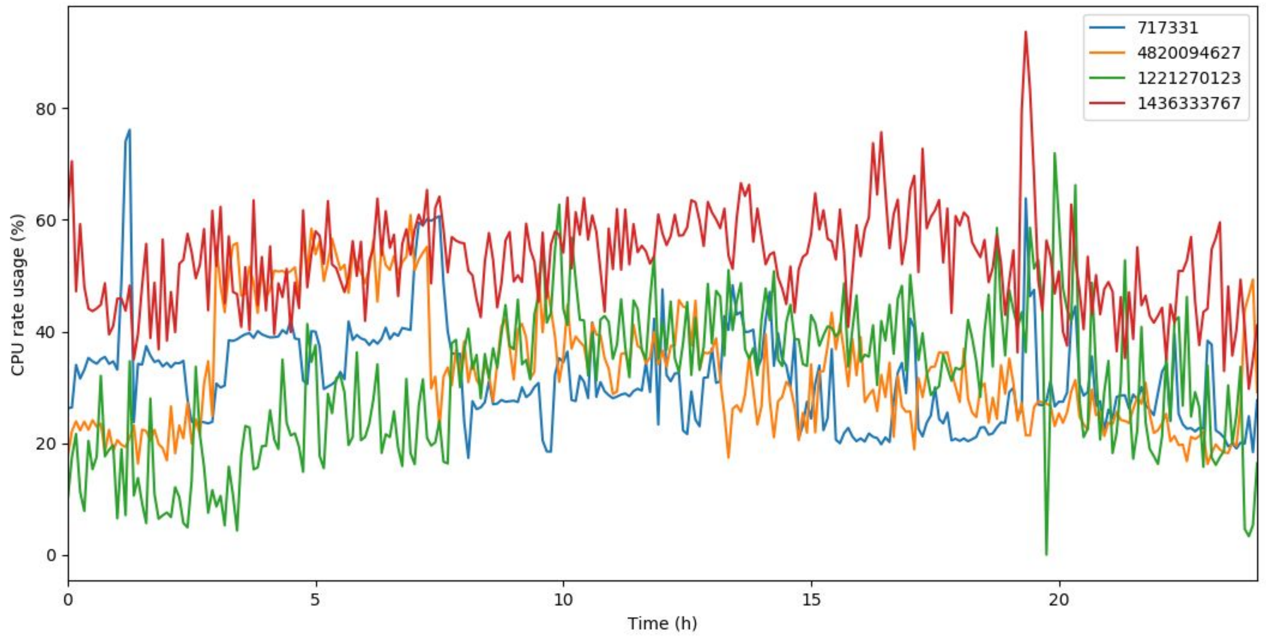
- Resource usage stability
- Job durations
- Repeated job schedule patterns
- Machine attributes

Python Analyzer

- Built a script to assist with extensive analysis of trace
- Collecting data samples from sections of the trace
  - Sampled from starting timestamp with specified duration
  - Randomly sample a number of machines to
  - 5 minute sampling intervals of job resource usage of selected machines
- Produce statistics and graphs for analytical purposes

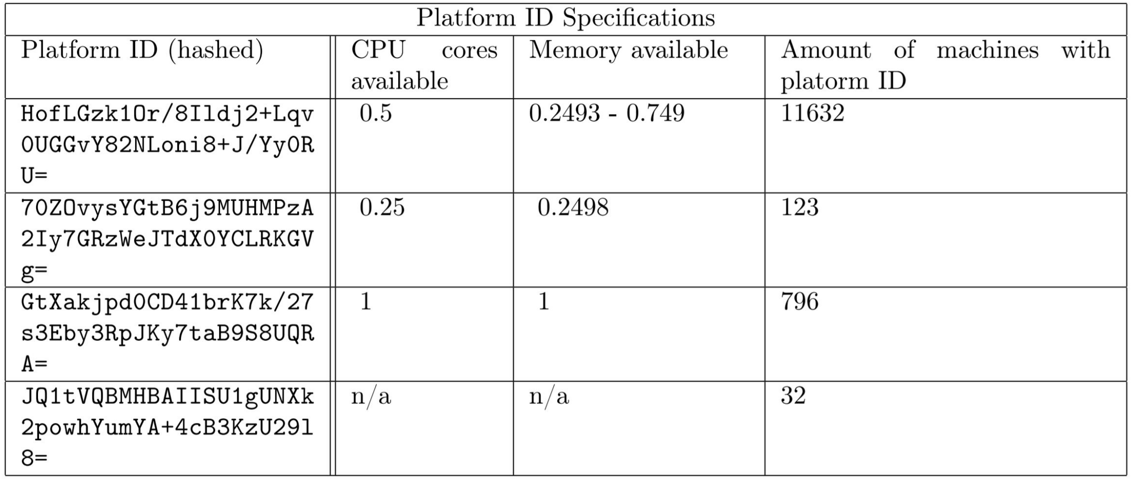
Python Analyzer

Example visualisation of 4 different machine workload timeseries data



Platform ID Analysis

- Machines grouped by 4 different platform IDs
- Describes machine builds and specifications

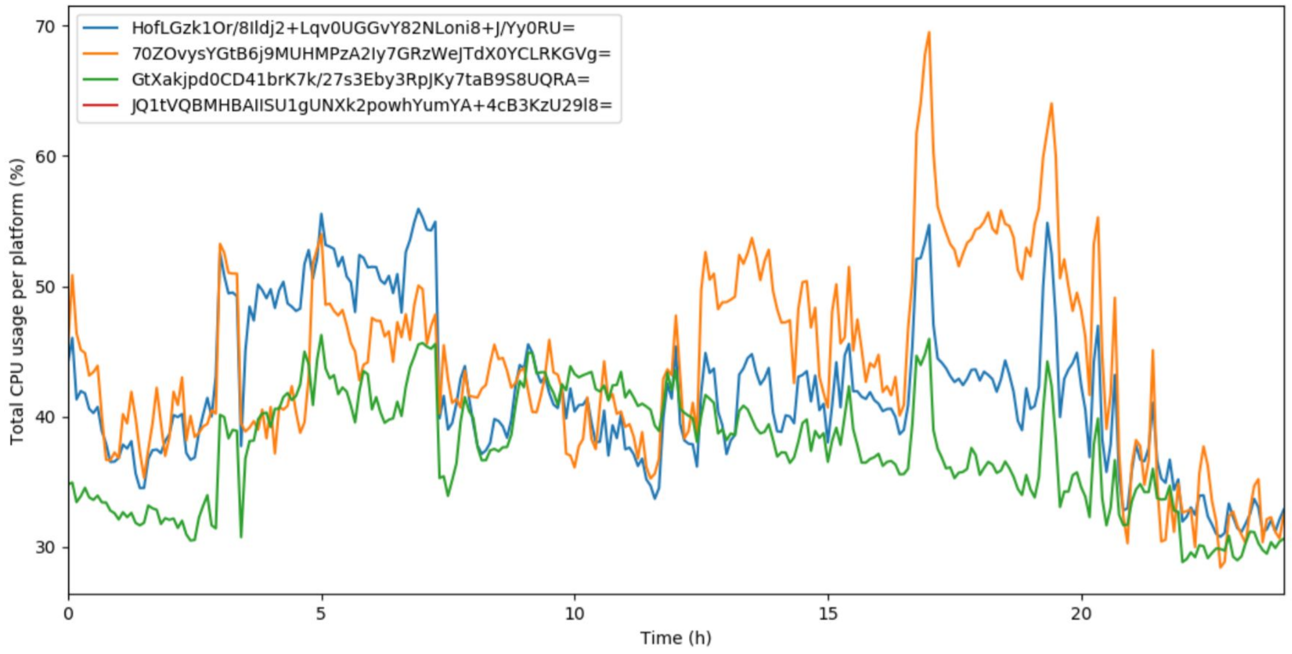


Platform ID Specifications

Platform ID (hashed)	CPU cores available	Memory available	Amount of machines with platorm ID
HofLGzk1Or/8Ildj2+Lqv0UGGvY82NLoni8+J/YyORU=	0.5	0.2493 - 0.749	11632
70Z0vysYGtB6j9MUHMPzA2Iy7GRzWeJTdX0YCLRKGVg=	0.25	0.2498	123
GtXakjpd0CD41brK7k/27s3Eby3RpJKy7taB9S8UQRA=	1	1	796
JQ1tVQBMHBAIISU1gUNXk2powhYumYA+4cB3KzU29l8=	n/a	n/a	32

Platform ID Analysis

Visualisation of overall machine usage grouped by their platform ID



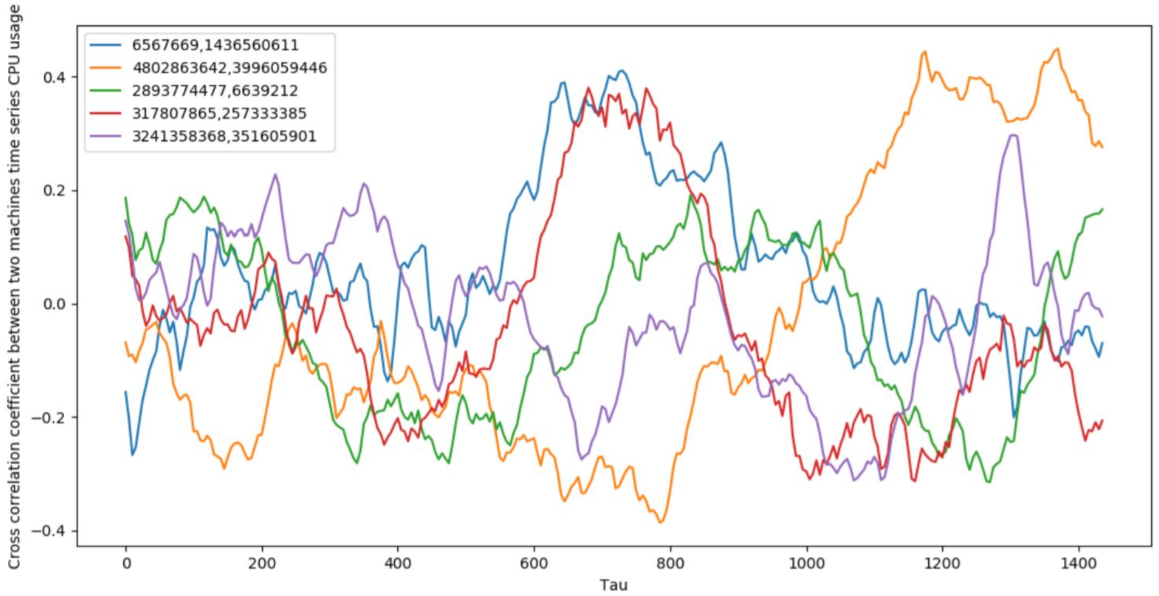
Correlation Analysis

- Analyse the relationship between machine subsets
- Evaluate the strength of the relationship using correlation coefficient

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2(y_i - \overline{y})^2}}$$

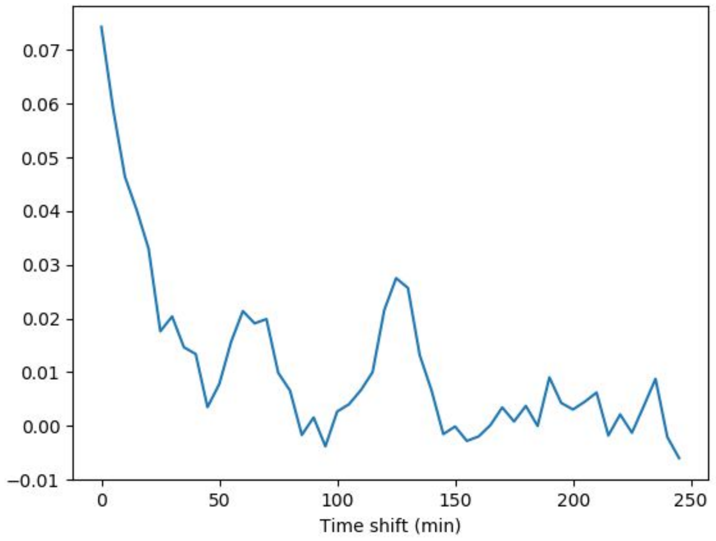
Correlation Analysis

Example of correlation analysis to analyse the extent of spatial correlation between a pair of machine workloads



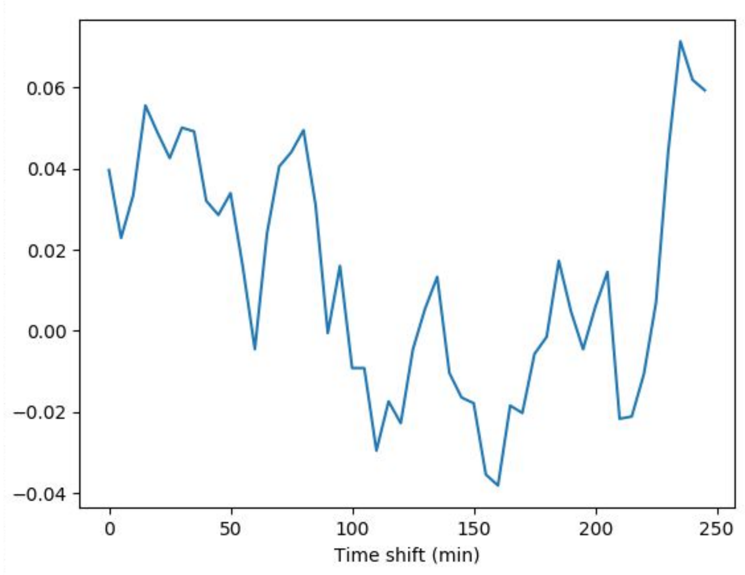
Spatial Correlation Analysis

Cross-correlation between pairs of machine timeseries, averaged across 500 random different pairs



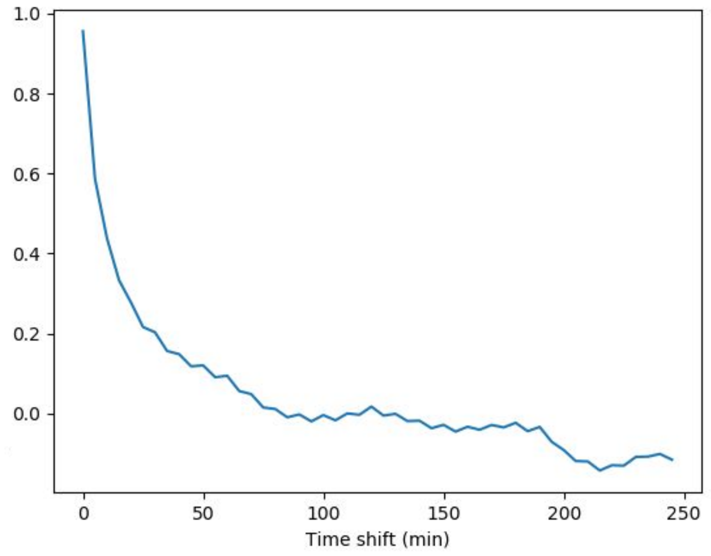
Spatial Correlation Analysis

Correlation between pairs of machine timeseries, averaged across 50 different pairs from sampled from specific platform ID



Temporal Correlation Analysis

Autocorrelation within an individual machine timeseries, averaged across 1000 different machines



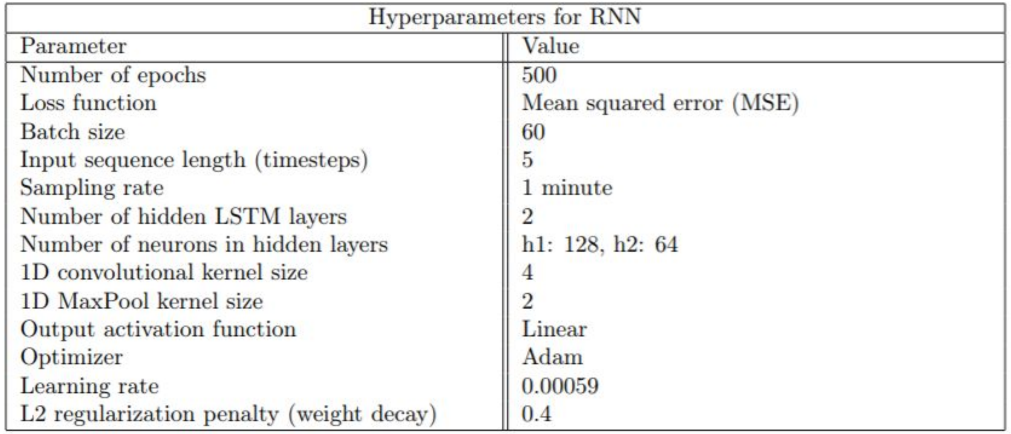
Trace Analysis Conclusions

- Overall weak spatial correlation between machine pairs timeseries despite machine attribute characteristics
- Difficult to characterise machine subgroups due to complexity of trace
- Strong temporal correlation within individual machine timeseries

Forecasting Model

Neural Network Model Overview

- Implementation of recurrent neural network model which includes:
  - 1D Convolutional layer
  - LSTM unit



Hyperparameters for RNN

Parameter	Value
Number of epochs	500
Loss function	Mean squared error (MSE)
Batch size	60
Input sequence length (timesteps)	5
Sampling rate	1 minute
Number of hidden LSTM layers	2
Number of neurons in hidden layers	h1: 128, h2: 64
1D convolutional kernel size	4
1D MaxPool kernel size	2
Output activation function	Linear
Optimizer	Adam
Learning rate	0.00059
L2 regularization penalty (weight decay)	0.4

Datasets

Compiled data from Python analyzer, sampled 24 hours of machine workload series data within the cluster trace from

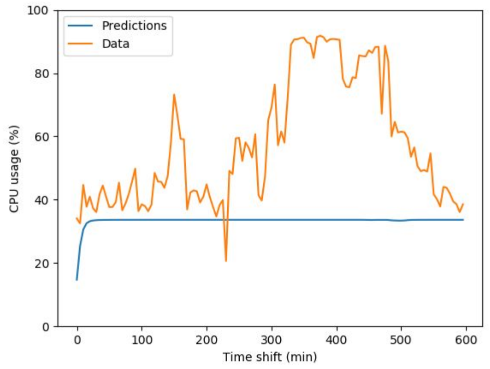
- **Training set** - sampled from 12 hours of data
- **Validation set** - sampled from 2 hours of data

Hyperparameter Tuning Approach

- Tuning and testing on one set of workload samples from **one specific machine** workload
- Increased sampling rate of dataset (1 min intervals)
- Batch training samples
- Experimentation of various different hyperparameter values

Data Preprocessing for Time Series Features

- Each sample fed into the LSTM requires a set sequence of timesteps
- The model uses the sequence as learning features to predict values
- Use sequence length as a model hyperparameter that can be varied



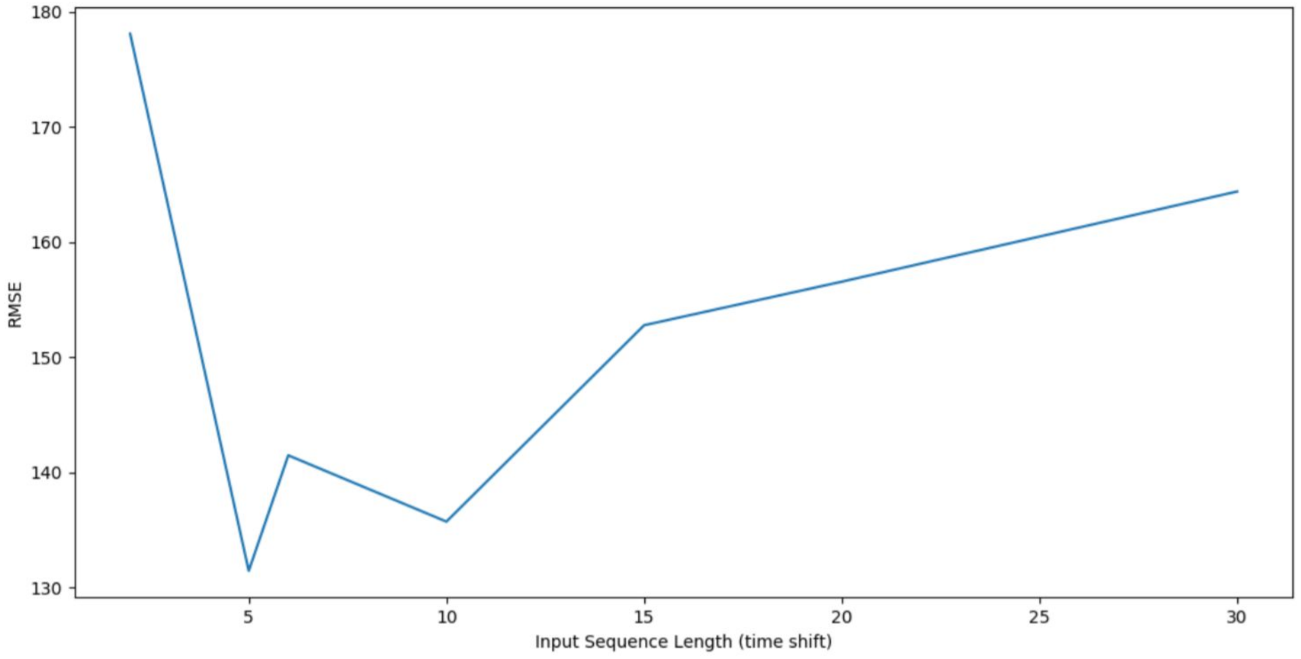
Data Preprocessing for Time Series Features

Example of modification of dataset [23, 55, 81, 93, 89, 82, 63, 47, ...] with timestep sequence length of 3

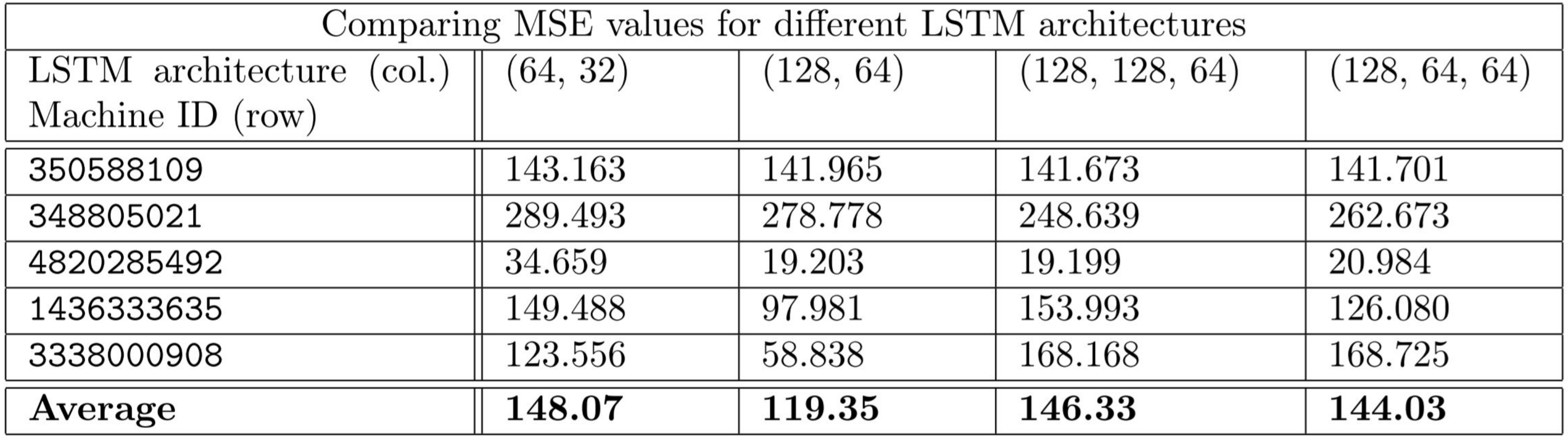
$$\begin{bmatrix} 23 & 55 & 81 \\ 55 & 81 & 93 \\ 81 & 93 & 89 \\ 93 & 89 & 82 \\ 89 & 82 & 63 \end{bmatrix}$$

Output targets of [93, 89, 82, 63, 47]

Timestep Feature Length



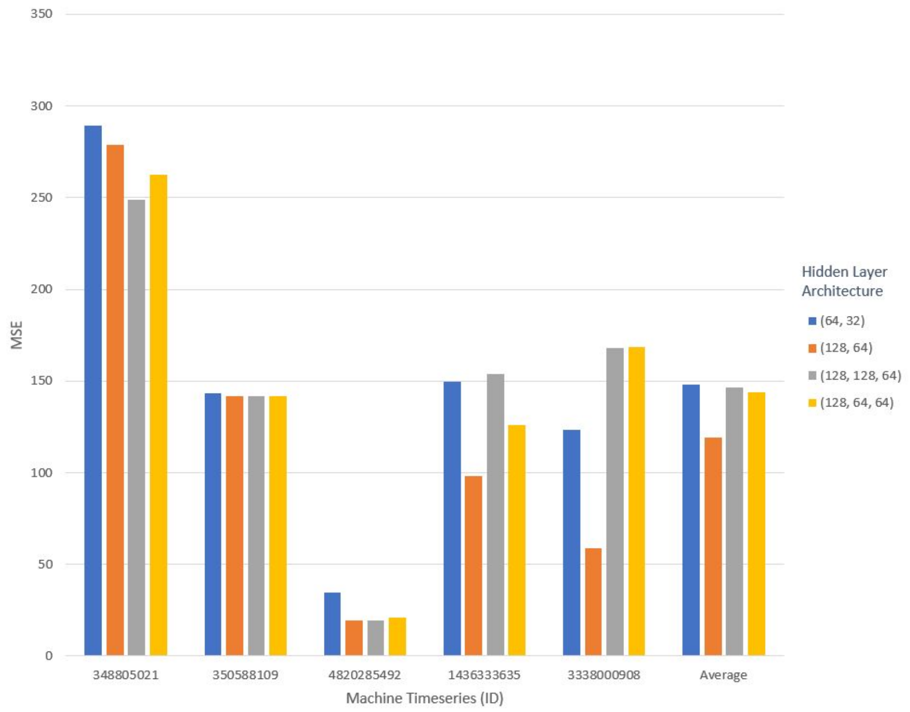
LSTM architecture



Comparing MSE values for different LSTM architectures

LSTM architecture (col.) Machine ID (row)	(64, 32)	(128, 64)	(128, 128, 64)	(128, 64, 64)
350588109	143.163	141.965	141.673	141.701
348805021	289.493	278.778	248.639	262.673
4820285492	34.659	19.203	19.199	20.984
1436333635	149.488	97.981	153.993	126.080
3338000908	123.556	58.838	168.168	168.725
Average	**148.07**	**119.35**	**146.33**	**144.03**

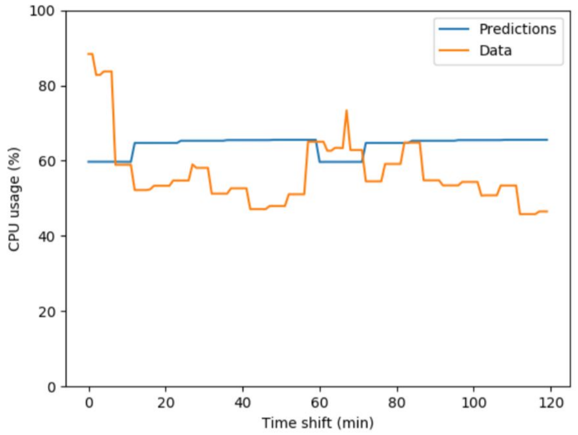
LSTM architecture



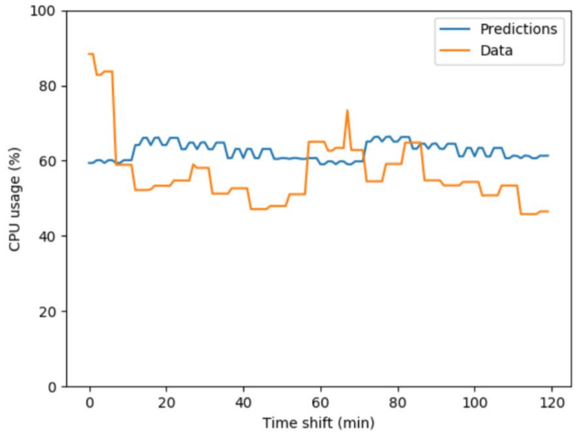
Visualisation of validation results over different LSTM architectures

1D Convolutional Layer

Can assist with processing a representation for complex scheduling features of machine workload timeseries



(a) Without 1D conv. layer, MSE = 142.088



(b) With 1D conv. layer, MSE = 119.557

Evaluation

Testing Dataset

- Trained on initial training dataset
- Collected from a different starting timestamp but on the same machine workload series
- Predicting future resource usage on unseen data
- Time sample period of 24 hours

Evaluation Metrics

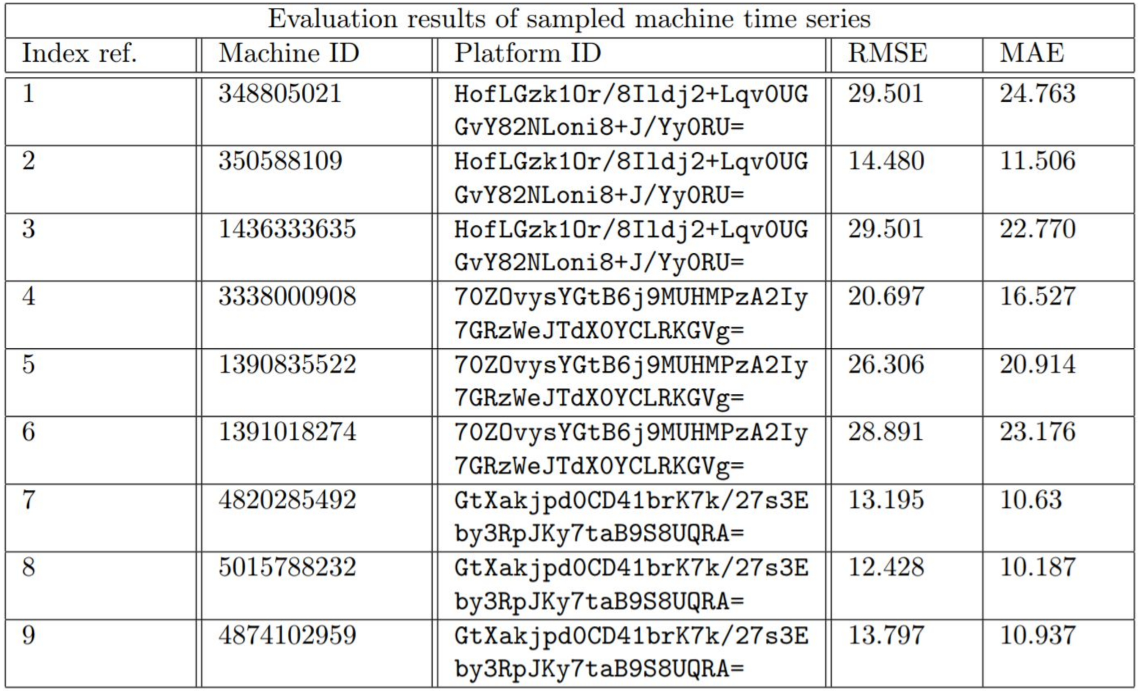
- Root Mean Squared Error (RMSE)

$$\sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

- Mean Absolute Error (MAE)

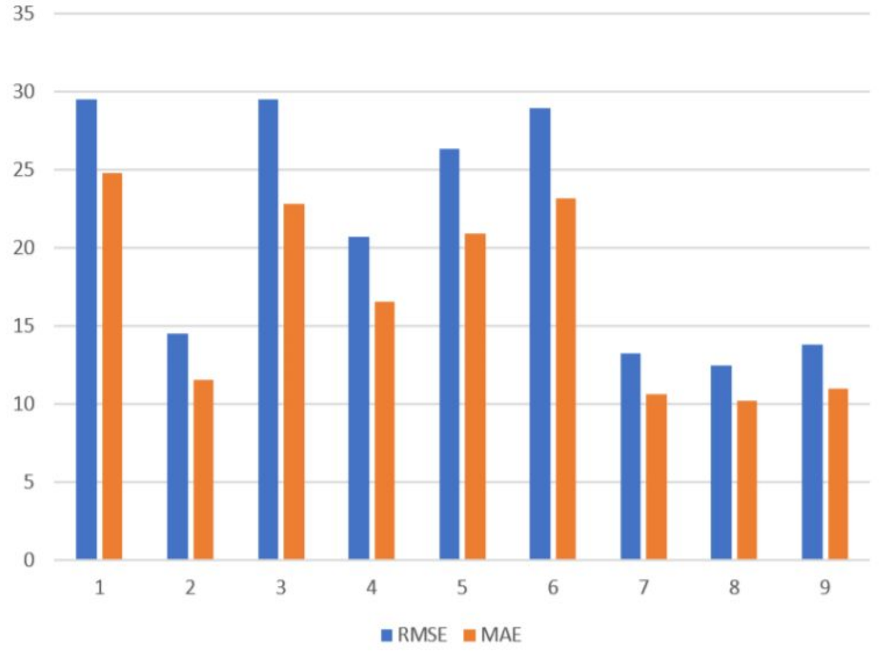
$$\frac{1}{n}\Sigma_{i=1}^{n}|y_i - \hat{y}_i|$$

Results



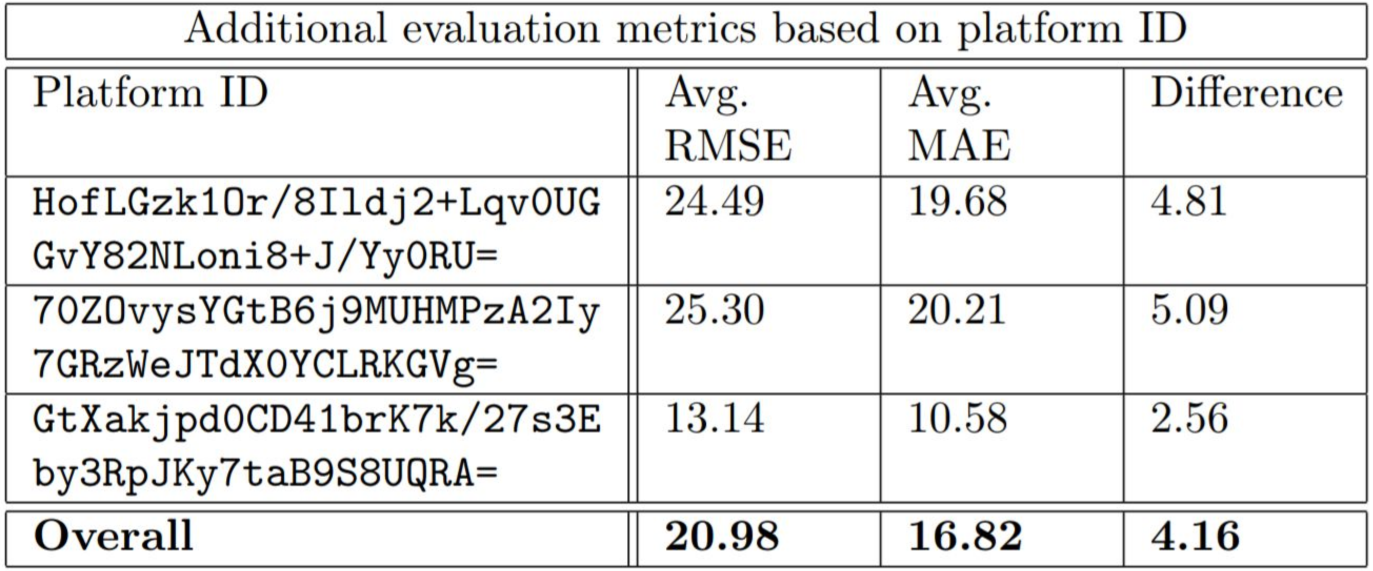
Evaluation results of sampled machine time series				
Index ref.	Machine ID	Platform ID	RMSE	MAE
1	348805021	HofLGzk1Or/8Ildj2+Lqv0UG GvY82NLoni8+J/Yy0RU=	29.501	24.763
2	350588109	HofLGzk1Or/8Ildj2+Lqv0UG GvY82NLoni8+J/Yy0RU=	14.480	11.506
3	1436333635	HofLGzk1Or/8Ildj2+Lqv0UG GvY82NLoni8+J/Yy0RU=	29.501	22.770
4	3338000908	7OZOvysYGtB6j9MUHMPzA2Iy 7GRzWeJTdX0YCLRKGVg=	20.697	16.527
5	1390835522	7OZOvysYGtB6j9MUHMPzA2Iy 7GRzWeJTdX0YCLRKGVg=	26.306	20.914
6	1391018274	7OZOvysYGtB6j9MUHMPzA2Iy 7GRzWeJTdX0YCLRKGVg=	28.891	23.176
7	4820285492	GtXakjpd0CD41brK7k/27s3E by3RpJKy7taB9S8UQRA=	13.195	10.63
8	5015788232	GtXakjpd0CD41brK7k/27s3E by3RpJKy7taB9S8UQRA=	12.428	10.187
9	4874102959	GtXakjpd0CD41brK7k/27s3E by3RpJKy7taB9S8UQRA=	13.797	10.937

Results



Visualisation of results for machine workload predictions

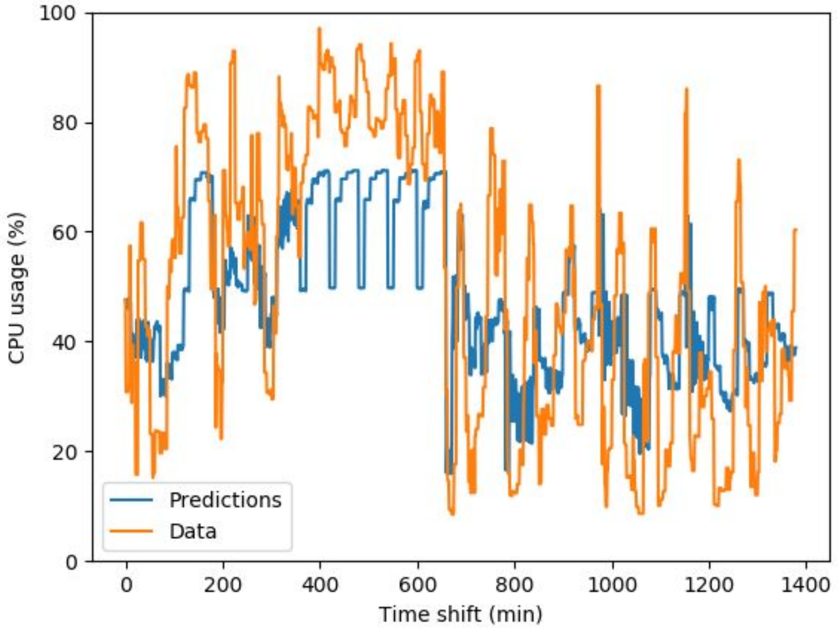
Results



Additional evaluation metrics based on platform ID

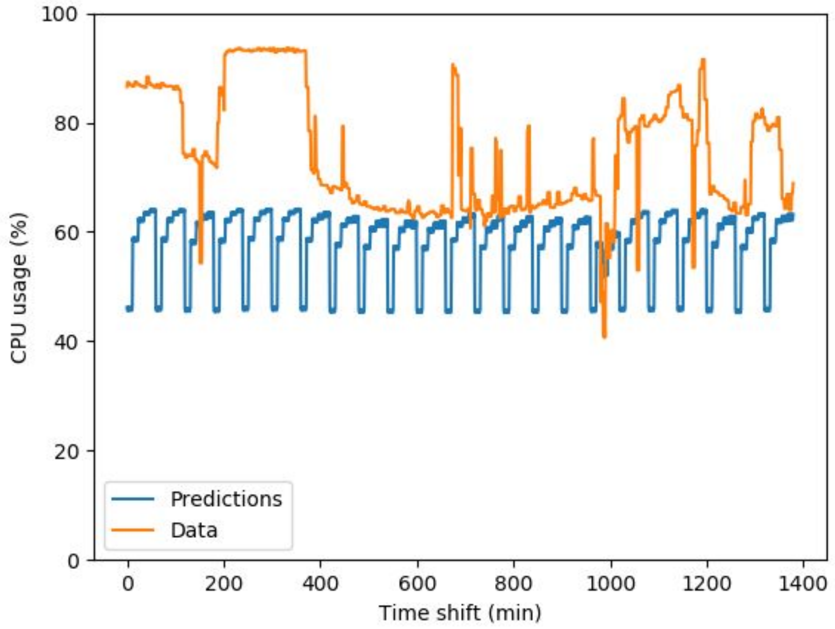
Platform ID	Avg. RMSE	Avg. MAE	Difference
HofLGzk1Or/8Ildj2+Lqv0UG GvY82NLoni8+J/Yy0RU=	24.49	19.68	4.81
70Z0vysYGtB6j9MUHMPzA2Iy 7GRzWeJTdX0YCLRKGVg=	25.30	20.21	5.09
GtXakjpd0CD41brK7k/27s3E by3RpJKy7taB9S8UQRA=	13.14	10.58	2.56
Overall	**20.98**	**16.82**	**4.16**

Evaluating forecasting results



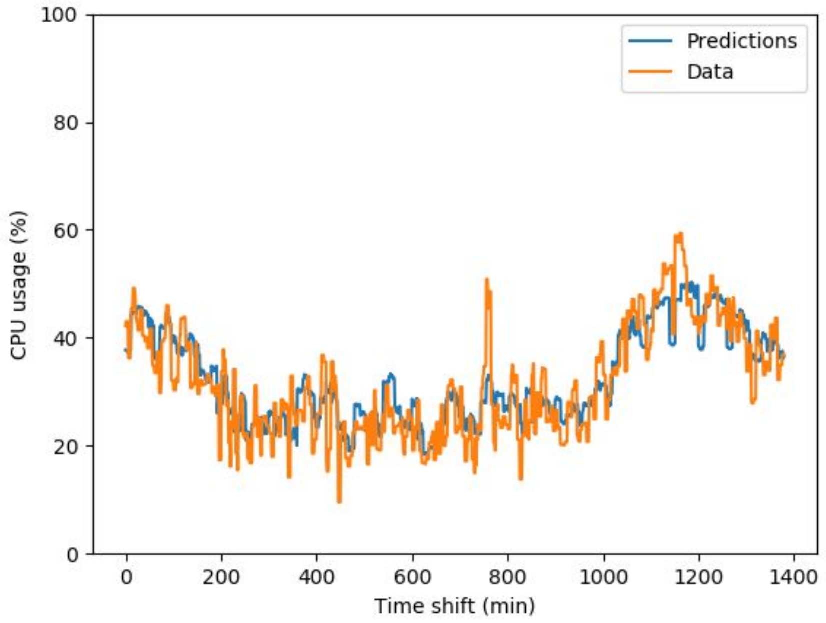
Machine ID: 348805021, RMSE: 29.501, MAE: 24.763

Evaluating forecasting results



Machine ID: 3338000908, RMSE: 20.697, MAE: 16.527

Evaluating forecasting results



Machine ID: 4820285492, RMSE: 13.195, MAE: 10.63

Conclusions from results

- Model generally performs well over the tested machine workloads
- Performs better on machines from platform ID with highest specifications
- Difficult time adapting to spikes and fluctuations within workload series

Further Improvements

- Prediction with live results to improve our forecasting model
- Use of grouped machine workload timeseries data to train new forecasting model
- Prediction of other cluster usage statistics such as machine failure likelihood at a certain time

Thank you for listening.

Any questions?