UMEÅ UNIVERSITY

Analysis and Reduction of Computer Performance Metric Collection for Predictive Analysis

A study of computer performance metrics predictive capabilities within a cloud data center.

Alban Gashi

Alban Gashi
Spring 2024
Degree Project in Computing Science and Engineering, 30 credits
Supervisor: Jerry Eriksson
Extern Supervisor: Torgny Holmberg
Examiner: Henrik Björklund
Master of Science Programme in Computing Science and Engineering, 300 credits

Abstract

This study, conducted in collaboration with Ericsson Research, explores the potential of utilizing metric data for predictive analytics within IT operations. The primary objective is to address underutilized data by investigating its utility in forecasting future trends and behaviors. The research is driven by two key questions: to what extent can metric data inform predictive behaviors and the identification of specific metrics most valuable for predictive analysis? The study focuses on three main aims: evaluating the quality and predictive suitability of Zabbix-collected data, assessing the strength of correlations within the datasets using industry-standard analytical techniques, and developing an inference model based on identified metrics. Initial findings indicate that while the metric data holds significant potential for predictive analytics, it exhibits high individuality among hosts, requiring careful feature selection and temporal resolution analysis. This research lays the groundwork for future studies to utilze datasets at Ericsson Research.

Keywords: Data analysis, Predictive analysis, Metric Data, Data exploration

Acknowledgements

Thank you, Torgny Holmberg, at Ericsson Research, for being an external supervisor and helping me see through the project. The contributions with weekly meetings helped push me in different directions and helped me know what path to take while exploring the options for giving the metric data characteristics.

Contents

1 Introduction

In the contemporary landscape of industrial standards, data centers are mandated to achieve unwavering operational continuity, striving for uninterrupted service 365 days a year, around the clock. This relentless pursuit of uptime is seamlessly supported by advanced monitoring tools that perpetually capture an array of metrics and log data from operational machinery. Such rigorous data acquisition culminates in generating voluminous datasets rich with intricate details about machine performance metrics and operational dynamics. This scenario is emblematic of the big data paradigm, characterized by its substantial volume and complexity. It poses formidable challenges in data analysis due to the overwhelming scale of the datasets involved. The sheer quantity of data collected in a single day can easily surpass the analytical capabilities of individuals, thereby necessitating the adoption of automated tools and methodologies for efficient data management and interpretation.
The narrative thus shifts towards exploring established techniques aimed at the analysis and definitive answer to metric data. Given its predominantly numerical nature, devoid of contextual cues, handled as a black box, the endeavor is to distill valuable insights from these data. The objective involves forecasting computer performance metrics. Giving a basis as the foundational bricks to build upon can hopefully help transcend beyond traditional indicators such as the “red lamp” indicator.

Moreover, the diverse nature of metric data introduces complexity in discerning correlations among various metrics. These metrics are influenced by the alignment of hardware components, which varies according to the specific workload and objectives intended. This aspect of the research delves into the analytical challenges posed by the heterogeneity of metric data, aiming to unravel patterns and relationships that could inform more effective and predictive management strategies for the industrial data center. By exploring tools to find valuable insights to help understand the contextual information of metrics of value. This study uses real, never-before-seen industry data and strives for answers.

1.1 Project Aim

This study represents a pioneering case study in collaboration with Ericsson Research, focusing on the practical application of predictive analytics within IT operations. By leveraging Zabbix-collected metric data, the aim is to address the underutilization of vast datasets in predicting future trends and behaviors. In partnership with Ericsson Research in Lund, this project delves into the untapped potential of metric data, investigating its utility and the extent to which it can be leveraged for predictive analytics. The impetus for this study stems from the current predicament where an overwhelming amount of collected data remains largely unanalyzed. This scenario presents a significant opportunity to sift through this data for valuable insights and ascertain the feasibility of employing predictive analytics

to forecast future trends or behaviors based on this data. This research employs established analytical tools to assess the quality and predictive utility of metric data, to enhance operational decision-making processes in a real-world industrial context.

This endeavor is envisioned as an industrial case study, utilizing actual data from Ericsson Research to conduct a comprehensive analysis. The project aims to uncover the potential of metric data in forecasting future trends, identifying patterns, and contributing to decision-making processes through predictive insights. Identifying which metrics have the most significant impact and how they can eventually be utilized in predictive models is crucial. The project is structured around critical inquiries concerning the insights and relevance of the metric data. The following research questions lead the project:

- To what extent can metric data inform predictive behaviors, and what value does it have?
- Which specific metrics are most valuable for predictive analysis from the collected data?

The aims of this project serve as guidelines to systematically address these research questions. By structuring the study around these aims, the project ensures a thorough exploration of the data, leading to actionable insights and practical applications within the context of IT operations at Ericsson Research. The collaboration with Ericsson Research provides a unique opportunity to explore these questions within a real-world context, leveraging actual data to validate findings and recommendations. Through this study, the project aspires to contribute valuable insights into the effective use of metric data in predictive analytics, offering potential strategies for Ericsson Research to enhance their forecasting capabilities. To methodologically give answers to the research questions, they are split into the following three aims:

Aim 1 Assess the quality and predictive suitability of Zabbix-collected metric data.

Aim 2 Assess predictive capabilities with well-established industry analytical techniques for predictive feasibility.

Aim 3 Develop and validate an inference model using identified metrics and methods.

1.2 Time plan

The project has an outlining time plan that is used to keep up with achieving work til deadlines; see Figure 1.

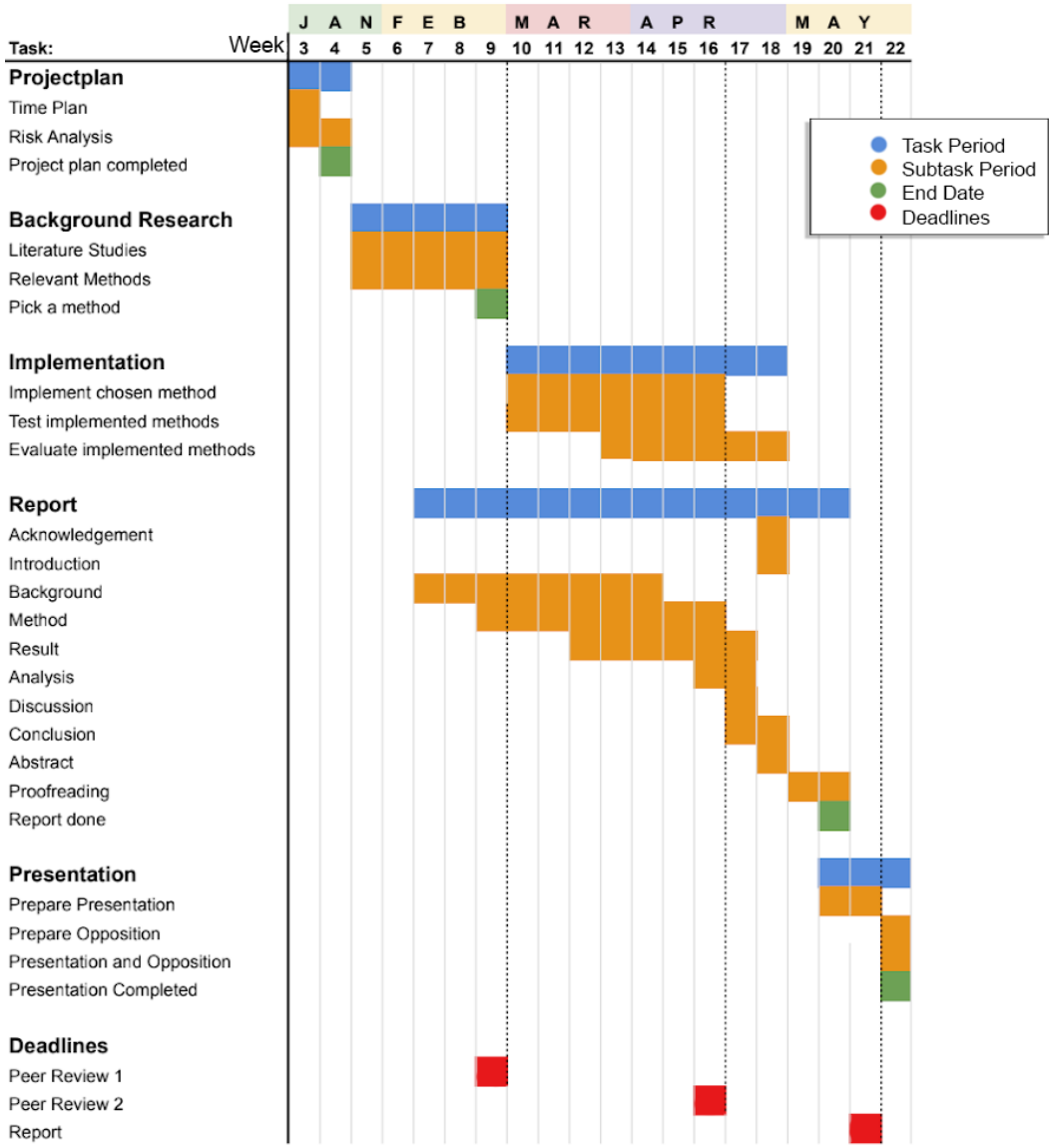


Figure 1: Gantt chart showing the project time plan.

1.3 Risk Register

In the case of this project, some risks are identified; see Figure 2 for included risk mitigations.

Risk	Motivation	Likelihood (L/M/H)	Impact (L/M/H)	Mitigation
Overly Broad or Narrow Scope	A topic too broad lacks focus, while too narrow may lack sufficient content.	L	M	Discuss tutors at the University as well as Ericsson for guidance on making it more compact
Originality and Relevance	Risk of choosing a topic that lacks originality or current academic relevance.	M	M	Revisit the objectives and see how I could add originality to it. Look at it from different angles and rewrite objective
Sickness	Becoming sick for an extended period of time	L	H	Reprioritize objectives within project to meet timeplans
Hardware failure	Loss of data, loss of notes	L	M	Will use several backups, external harddrives as well as cloud storage options
Will only collecting data solve problem	The problem might not originally just be with collecting metric data to reduce general data	M	M	Other alternatives can come up
If supervisor quits at Ericsson (Torgny)	Might get tired of his job.	L	M	Have to find a new tutor to help with.
Ericsson doesn't have the appropriate data	The data might not give results	M	L	A result will still be given to a give a final conclusion.

Figure 2: Explaining the risk and mitigations during the course of the project.

2 Background

This section covers the background knowledge and definitions useful for understanding the project's specifics and definitions.

2.1 Data Center

Different cloud data centers exist, ranging from public to hybrid and private. They store information to help run several companies' businesses. However, they can generally be seen as physical buildings that house IT infrastructure for running and developing applications or services. As data centers house servers and services, they can be used by companies and private persons since they also need effective management[15]. The data center holds different machines known as hosts. They can be utilized by several persons operating with different workloads. As people do not have the same tasks at Ericsson, these hosts create a heterogeneity of metrics in activity. The data center is also shared by people worldwide that operate on the servers.

Metrics are a form of quantitative measurement that can help affect a company's decision-making [8]. The metric in the context of this study is computer performance metrics. The metrics are then measurements of the usage of, e.g., CPU utilization. Zabbix is a software for monitoring the infrastructure by collecting said metrics [45]. Ericsson Research has used the tool to collect several months of data on their servers, measuring their performance every minute, illustrated in Figure 3.

There are some considerations to be had when looking at the different types of metrics. The metric type that this study analyses is computer performance metrics; a good or significant metric would have to be interesting and show the history of the cases being looked into. The metric needs to be reliable for prediction. There can also be five ways of considering the characteristics of a good one: *i*) does the metric hold relevance for the context of the situation, *ii*) can the metric being measured, *iii*) is the metric actionable, *iv*) is the metric robust and *v*) is it readable [29]?

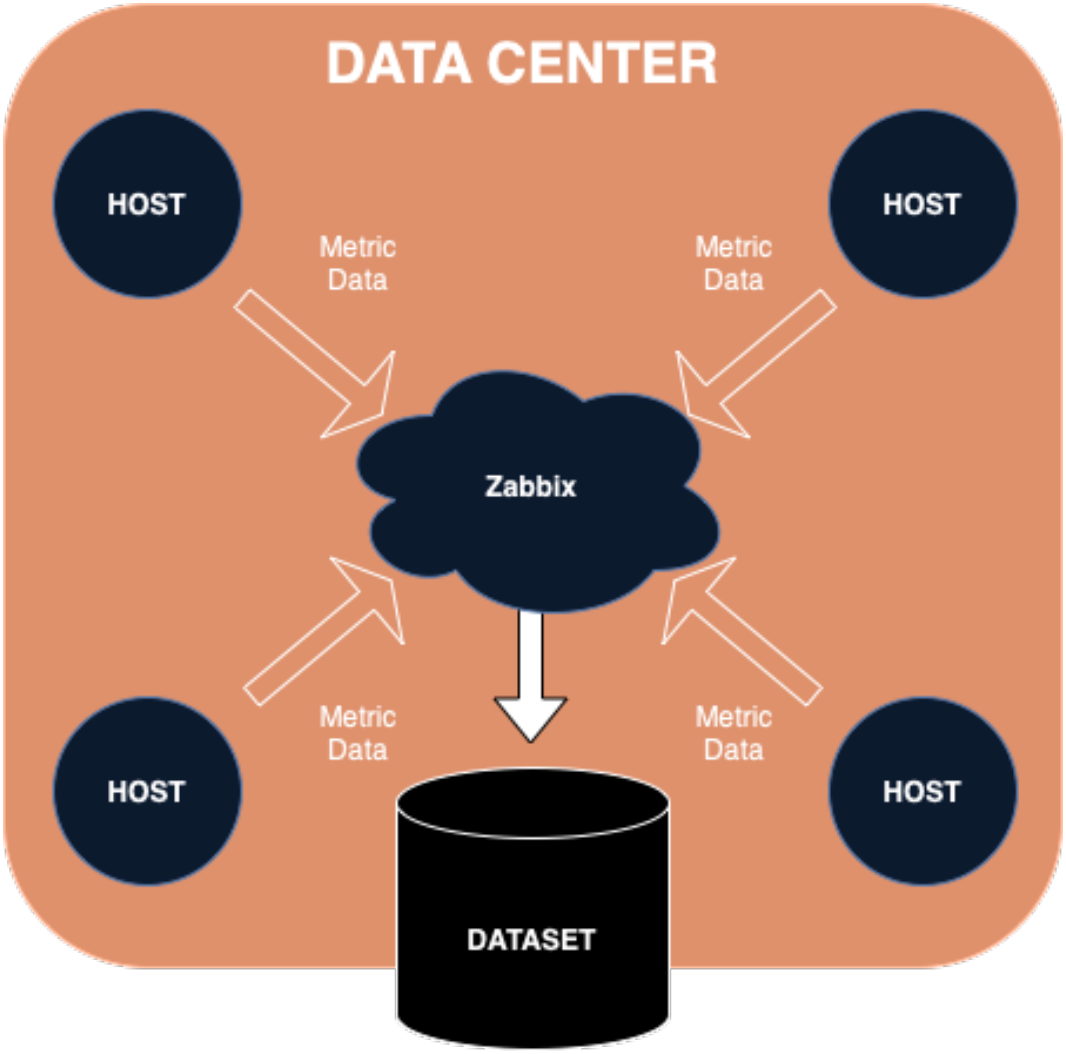


Figure 3: How the data center dynamic works between hosts. Hosts are within the data center, and the Zabbix monitoring tool collects the metrics.

2.2 Artificial Intelligence

Artificial Intelligence (AI) is the simulation of human intelligence in machines programmed to think and learn like humans. It encompasses various subfields, including machine learning, where algorithms improve through experience, and deep learning, which involves neural networks with many layers. AI applications range from natural language processing to autonomous vehicles, significantly impacting various industries by enhancing efficiency and enabling new capabilities[16].

2.3 Big Data and Big Data Analytics

Big data is typically characterized by three main components: volume, velocity, and variety. Volume refers to the sheer amount of data within a dataset. Velocity denotes the speed at

which data is received and processed. Variety pertains to the diverse types and sources of data, highlighting its unpredictability. In recent years, two additional 'Vs' have been recognized: value and veracity. Value addresses the intrinsic worth of data once its purpose is identified, while veracity relates to the accuracy and trustworthiness of the data. Big data encompasses vast datasets that can be either heterogeneous or homogeneous in nature [30].

Analyzing big data involves employing advanced techniques to manage these extensive, varied datasets. Due to their size and complexity, these datasets require processing by modern technologies such as AI. The goal of big data analytics is to improve modeling, predict future trends, and facilitate more intelligent and rapid decision-making [13]. Predictive analytics, a subfield of advanced analytics, uses historical data to forecast future outcomes. This approach is highly valuable for strategic decision-making across various industries, from marketing and sales to IT.

A specific application of this is Artificial Intelligence for IT Operations (AIOps), which leverages AI to automate and streamline IT service management and operational workflows. By training on big data and utilizing machine learning, AIOps enables IT teams to respond more swiftly to downtimes and shutdowns, allowing them to act proactively [17].

2.4 Machine learning

As mentioned earlier, Machine Learning (ML) is a subfield of Artificial Intelligence that aims to imitate the human way of learning algorithms and data to improve the accuracy of the machine. It is generally used to create predictions or classifications based on input data, which can be labeled or unlabeled. Labeled data is data that has been previously classified or characterized. Estimations are created based on the data and previous data points. It is a model that is usually trained with previous data and then estimates[18]. There are two basic approaches to AI in the industry: unsupervised and supervised learning. The difference is that unlabeled data is used for unsupervised learning, while labeled data is used for supervised learning. The supervised approach guides the algorithm in predicting outcomes and classifying data by inputs and outputs, making learning more useful over time. The unsupervised way discovers hidden patterns in the data with the algorithm, which requires no human intervention or guidance for getting an outcome [14].

A commonly used Machine Learning algorithm is Random Forest[19]. By combining several decision trees, an output is given. The algorithm asks itself questions such as "Should I...?" before it arrives at a final decision known as a leaf. Three main hyperparameters are set before training: host size, number of trees, and number of features sampled. By configuring these parameters, the random forest model can give vastly different results. Regression algorithms[41] are used to make predictions based on historical data points. The same principle can be applied with Random Forest, a Random Forest Regression algorithm, to solve various regression problems.

2.5 Correlation Analysis and PCA

Correlation analysis involves examining the relationships between variables to determine how closely connected they are. This project uses correlation analysis to identify patterns

and draw conclusions about data characteristics. It quantifies the extent to which two variables change in relation to each other.

Within this project's scope, correlation analysis is applied to understand the relationships within the metrics themselves. Two main statistical methods are used: the Autocorrelation Function (ACF) and the Partial Autocorrelation Function (PACF). ACF is a time-based function that defines the relationship between data points in a time series, assessing the similarity or correlation between these points at different times, known as lags. It quantifies the relationship between a variable's current and past values, considering the intermediate lags for the lag being analyzed [28]. On the other hand, PACF also examines the relationship between data points in a time series but removes the effect of intervening observations. PACF specifically measures the direct correlation between observations at a given lag without the influence of shorter lags, meaning that, for example, lag three indicates a direct correlation of three lag units [34].

Principal Component Analysis (PCA) is a linear dimensionality reduction technique. It is used for dimensionality reduction in high-dimensional datasets. It performs orthogonal transformation between variables, treating them as independent to develop new components. These components create new patterns based on the variance retained within the data, which gives great insights for feature selection to identify potential correlations. PCA helps reduce noise in data and is a useful technique used in the exploration of data where underlying data patterns may not be clear. Features with high variance are often important within a dataset. Implementing PCA in complex datasets can also reduce overfitting in machine learning models, ensuring the model does not only predict one type of data [12].

2.6 Frameworks and Tools

Pandas is a well-known Python library that offers numerous functions for managing large datasets. It provides a fast and efficient way to process and perform various operations on vast amounts of data, making it an industrial standard for data analysis [32]. Pandas is often used alongside Matplotlib, another popular Python library, which serves as a powerful visualization tool. Matplotlib enables the creation of a wide range of visualizations, from static charts to animated graphs, helping to effectively present data findings and enhance the understanding of underlying data patterns [26].

Scikit-learn is a popular and versatile machine-learning library for Python, widely utilized in academic research, prototyping, and production. It offers simple and efficient data mining and analysis tools, built on top of NumPy, SciPy, and Matplotlib. Scikit-learn includes a variety of machine learning algorithms, such as Random Forest, making it a comprehensive tool for various ML tasks. It also holds the function to perform PCA[33]. Additionally, Statsmodels is a Python library that provides statistical classes and functions. It is extensively used for data exploration, statistical testing, and validation. This project employs Statsmodels to apply ACF and PACF to the metric data [40].

3 Related Work

For data exploration, there are common approaches to handling it; preprocessing a big part is getting familiar with the data before modifying it. Handling duplicates and, removing null values, understanding the data type, it is important to have good data quality[24][36]. There is also a good advantage of being able to verify with the human eye through visualization of how the data is. Clarifying any mistakes or discovering patterns through human intuition might give huge insights into the data. It is a reason that even though there are a lot of automated frameworks, visual tools still exist [37].

Given the importance of assuring quality within the data used for data analysis [2], [38] given that poor data can hurt ML models. A survey found by comparing different data quality tools that [3] sets a baseline for judging the quality of the data. There is also the study[43] that explores different predictive performance metrics with a black-box approach. This study also has environment mapping, which collects a relation between the application and the computer resources to better understand the demand and overhead placed. Highlighting the effectiveness of non-intrusive, data-driven methods for predicting performance metrics. The study also highlights data-driven insights, where understanding past data gives leverage for more informed decision-making. This finding is also supported by previous studies when using data for predictions [22], [27], which states the importance of historical data for improving predictions.

Furthermore, studies like [10], show that a higher resolution can improve simulation accuracy, which suggests a better forecasting ability. Even more, the study on temporal resolutions [5] would suggest that a higher temporal resolution does yield better accuracy but not marginally depending on the domain of the prediction. The domains of the studies are different but also suggest that a more optimal temporal resolution for the domain can yield better predictions.

As this thesis assumes that a general model of prediction cannot be applied between heterogeneous hosts, there is the study [39] that also suggests greater accuracy in time-series data can be achieved by building separate models per unique user, that in this context translates to each host. As the metric data was collected, there is no previous knowledge of the correlations between the high dimensions. There is a need to tell what variables might hold greater importance. For machine learning models, studies such as [1] introduce the need for variable selection techniques that are less computationally expensive. The importance of the feature selection for improving the model is also suggested in [7], to the point that the correction variables can even improve performance when trained on sets with missing data [11].

The mentioned reports serve as a baseline for creating a solution. This study focuses on previously unstudied data collected by Zabbix. The thesis addresses the research gap of underutilized datasets at Ericsson Research for predictive analytics, enhancing operational decision-making in real-world industrial contexts.

4 Solution design

This chapter presents the solution design for the aims by explaining the tech framework and statistical tools and setting up a high-level architecture. Each set aim has two objectives as well as the theory framework to get results. As detailed in Section 1.1, the aims and objectives serve as essential navigational tools that direct the research efforts toward answering the research questions, ensuring that all aims are aligned with these fundamental inquiries. The unsolved challenges with this project are the ones that Ericsson Research has yet to explore: the uncertainty of the data, whether it can be used for prediction, and if these criteria are filled, test it with a simple inference model. The importance of each step is motivated by the related works in Chapter 3.

4.1 Tools and dataset

The data analysis uses the Ericsson Research data portal, which also provides Jupyter Notebook [1], a web-based development service. Through this portal, the datasets can be accessed from anywhere but have the consequence of not keeping the data locally, where interrupts and limitations such as speed limitations with on-demand interaction with certain files. The Jupyter Notebook is a highly flexible platform for installing valuable frameworks and libraries to help with analysis. Pandas and Matplotlib make data easier to analyze and visualize, essential for all thesis aims to answer the Research Questions, see Section 1.1. Ericsson Research provides voluminous datasets containing months of data collected every minute by the Zabbix monitoring tool for each machine. The data collected are computer performance metrics.

4.2 The preprocessing of data

The initial step involves thoroughly processing and exploring the data, which includes understanding the metrics collected and their implications as defined by Zabbix documentation [44]. The data has never been thoroughly analyzed, so administering any analysis or description of the metrics gives insights into the nature. The problem is whether there are limitations to the metrics and whether the datasets collected are of quality. Therefore the objectives are as follows for Aim 1:

Objective 1.1 Evaluate the completeness, accuracy, timeliness, and consistency of metric data collected by Zabbix, analyzing its suitability for use in predictive modeling within an industrial setting.

[1] https://jupyter.org/

Objective 1.2 Identify and quantify the limitations and challenges in the data, such as missing values and noise levels, that may impact the effectiveness of the data set.

The study follows the four most common and generally accepted Metrics to assess the data quality. This also follows the International Organization for Standardization (ISO)[2], for Data Quality Model ISO/IEC 225012:2008 [20].

Accuracy is for measuring the magnitude of the error the data holds, calculated as follows:

$$\text{Accuracy} = \frac{\text{Number of Correct Data Entries}}{\text{Total Number of Data Entries}} \tag{4.1}$$

Completeness is the measurement, and this study's approach follows the Heinrich approach of counting true values [9]. This is calculated as follows:

$$\text{Completeness} = \frac{\text{Number of Non-Null Data Entries}}{\text{Total Number of Data Entries}} \tag{4.2}$$

Consistency is the measurement of the data integrity, which speaks for the data's reliability. In turn, the question is, does the data adhere to the same constraints between all hosts? This has to be a manual formulation of the original equation, which makes it more complicated and would measure up to a few things. Do all hosts capture metrics the same way? Do they all have the same ranges of data, and do they collect data at the same intervals? Consistency is calculated as follows:

$$\text{Consistency} = \frac{\text{Number of Consistent Data Entries}}{\text{Total Number of Data Entries}} \tag{4.3}$$

Timeliness is the measurement of the relevance of the data to the task at hand. Data has a timeline for use, which creates a decline in value. This is calculated as follows:

$$Q_{\omega}^{\text{Time}}(t) := \exp(-\text{decline}(A) \cdot t) \tag{4.4}$$

In this context, ω is the considered attribute value, and $\text{decline}(A)$ is the decline rate, which specifies the average number of attributes that become outdated within the time period t. This is calculated in compliance with a manual inspection of the data in combination with Ericsson Research domain knowledge, using the Pandas tool to clean up the data and search for patterns within the data from the manual inspection. This is supported by the use of Matplotlib for visual graphs.

The success of this aim would be that the data is ensured to be viable and data quality is assured. The metrics have variability, value, and relevance in history.

4.3 Evaluation of patterns and predictive strength

This is the second step to evaluating the integrity of the metrics for predictive capabilities. From the work set in Chapter 3, a correlation can be seen between having good feature selection for improving ML models. While this is true, strong historical data and data from the right temporal resolution can also help boost the forecast. The answer that lies in the data is answered using statistical methods. The objectives to verify this for Aim 2:

[2]https://www.iso.org/home.html

Objective 2.1 Investigate the data with underlying patterns and strength in feature correlation, assessing their influence on model efficiency and accuracy.

Objective 2.2 Analyze the efficacy of important metrics statistical tools in revealing temporal dependencies and their predictive power within the context of Zabbix metrics.

The use of PCA as a feature selection tool has many advantages. The reduced dimensionality helps mitigate overfitting a model. In keeping the original data's patterns and trends [25]. This makes it highly suitable for high-dimensional metric datasets. This makes it one of the most useful tools when training Machine Learning models as it is a slow process; it helps reduce the training data for faster models [23]. An advantage PCA contributes is that the information of the data still remains in the principal components with the most variance, even through dimensionality reduction. A beneficial way of maintaining historical accuracy while restraining dimensionality size. However, the cost can come with increased error rates. PCA can interpret the principal components' output with the original features, making it easier to see what features contribute the most variance to the global data structure; these give insights into what features influence the data. Standard scaling is recommended when comparing several features; PCA becomes highly sensitive data with higher variance when not relative to the same scale.

To find the most important features, ML techniques such as the Random Forest algorithm are employed further to select strongly correlated features within the complex dataset. The Random Forest algorithm, specifically the Random Forest Regression [3], is utilized in combination with a feature selection method called SelectFromModel, both of which are provided by the Scikit-learn library [4]. SelectFromModel identifies features that contribute significantly to the model's predictive performance by selecting those that lead to the purest leaves and improve decision-making in predictions. They have proven that Random Forest can be useful for regression problems and feature selection and is not sensitive to outliers or noisier data while still performing well [21]. This highly motivates the tool's usage as the metric data is prone to vary.

The statistical method is ACF, used to find strength in historical data and determine the best temporal resolution for better predictive capabilities. It has shown to be useful for judging historical data and its strength in prediction[42]. Therefore, it can be used to predict what metrics have a strong historical background and might be more suitable for prediction. It can also be used to see in what lag it finds the strongest cycles. Comparing hours and days is useful to see if it can find correlations towards itself in the metrics and in what time unit it finds the strongest lags. Combining this with a similar algorithm, PACF, nuances the findings and correlation and motivates stronger evaluations. While ACF considers the total correlation between an observation and its lag, PACF isolates the correlation contributed by the lag itself, excluding contributions from intermediate lags. They have also been employed to help strengthen neural network models for univariate time-series forecasting [6]. By applying a 95% confidence interval to the output lags, it can filter and identify the statistically significant lags, assuming they are important. This means that only the lags outside the confidence interval are considered significant and thus likely to contribute meaningful information to our analysis.

[3] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
[4] https://scikit-learn.org/stable/modules/feature_selection.html

If the data consistently reveals underlying patterns, rendering it optimal for feature selection, and if the metrics display robust historical data points conducive to forecasting, all complemented by a temporal resolution that enhances predictive effectiveness, then this would fulfill the criteria for success for the second aim.

4.4 Model creation based on previous successes

This section of the solution is based on evaluating and testing the metrics' efficacy for inference modeling. Therefore, a simple inference model and an evaluation of it can help. These are the two objectives for Aim 3:

Objective 3.1 Construct a baseline inference model utilizing the most predictive features and temporal patterns identified through *Aim 2* analyses.

Objective 3.2 Test the performance of the inference model against real industrial metrics to assess its accuracy and reliability, comparing metrics

The baseline inference model is used to judge at an early phase to what degree metrics can be predicted, which is used in combination with the features selected from the mentioned statistical tools in Section 4.3. The model is created using Random Forest Regression and evaluated using the mean square error for the data model to evaluate accuracy. This is done on a high level, where few to no modifications are not done. It is used as more contextual data to draw conclusions based on the existing metrics. The model utilizes the most useful features determined through previous evaluations in Section 4.3. A generally recommended amount of trees (trees are decision trees using different random data subsets to make predictions, and their combined output improves accuracy) is around 100[4][35] for a baseline prediction, and increasing the number of trees does not result in better accuracy [31]. Too many trees can result in overfitting or become marginal in accuracy.

Success for aim three would be any type of answer possible. Findings that show metrics used as variables in feature selection to predict other metrics can show predictions to any degree. This means correct or even non-correct predictions yield answers. This, in turn, means using error detection to test the accuracy of the predictions.

4.5 High level architecture

For the aims explained on a high level, see Figure 4.

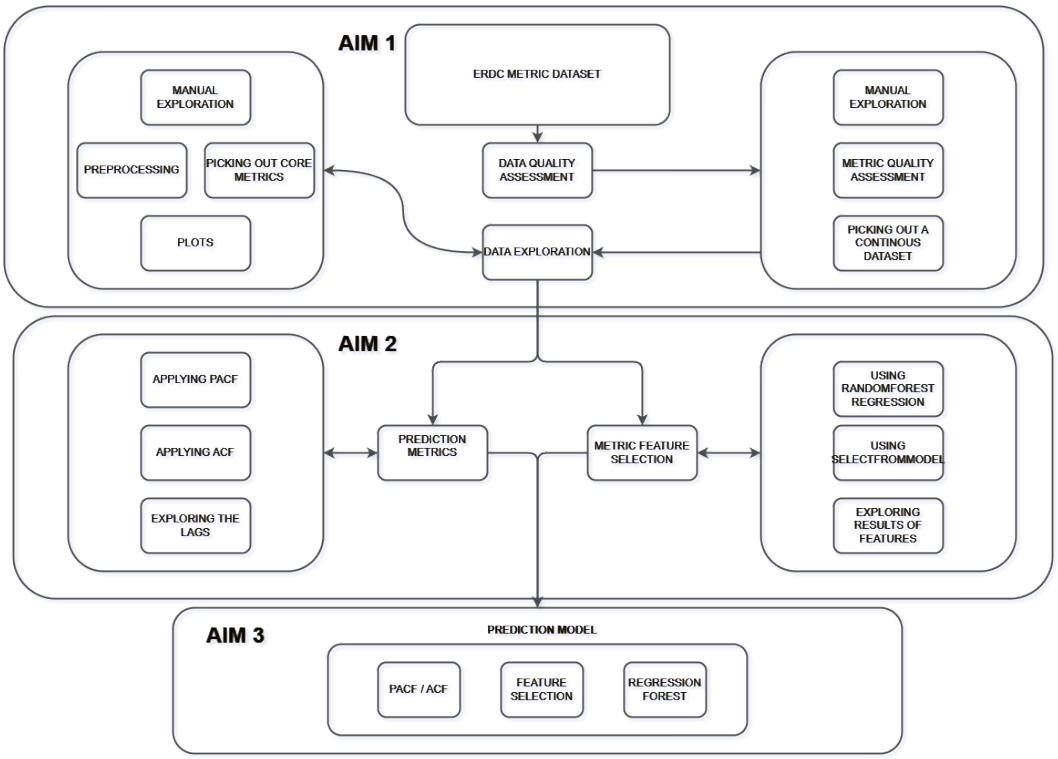


Figure 4: An overview of the aims on a high level.

5 Methodology

This chapter explains the proposed methodology and how to use the theories and solutions explained in Chapter 4. It navigates putting the solution to the methodology for each aim, as they seek to answer the research questions formulated in Section 1.1.

The first aim is to look through the data to establish ranges and units of the metrics. This is done with Jupyter Notebook, by building pipelines and applying similar preprocessing methods, clearing values and evaluating them between hosts. To find the definitions of the collected data, the metrics are compared with official Zabbix documentation[1]. This is combined with assessing the data quality of the whole data, as per the equations mentioned; see Section 4.2. The equations of *Timeliness* (4.4), *Completeness* (4.2) and *Accuracy* (4.1) remains the same. The equation for *Consistency* (4.3) is adapted for three measurements: the consistency of units, metrics (ones all hosts share), and time of collection between hosts.

The data quality then follows pursuit in exploration to find continuous data to create a dataset that is as close to 100% in data quality for analysis. The exploration of the datasets is done mainly by using the tool Pandas to explore the datasets and using Matplotlib to visualize the metrics, i.e., using line plots, scatter plots, and diagrams to see and understand the structure of the data. After understanding what is within the different columns and what is connected, cleaning up the data would be necessary. Looking at the different plots, trying to establish any patterns that might be obvious. This is in combination with exploring the data and establishing if there are more valuable periods where continuous data exists; this, in turn, can give better results and support for the statistical methods used after establishing a continuous time period. The data cleaning mainly removes metrics that were not collected. The outliers were also not removed, as they are argued to show eventual patterns or trends of the user workload.

All the following methods are applied to an established continuous data quality-assessed dataset from the results obtained through the data exploration. From the Statsmodels module, ACF and PACF are the statistical methods used to judge historical strength in data. Random forest regression (with SelectFromModel) from the SciKit library is used to judge the best feature selection with machine learning. PCA is applied to see variance, give further insight into the underlying patterns, and reduce noise. It is from the SciKit library.

The results are then printed and compared to see what period of time offers the most significant trends and cycles for each metric. The visual plots and assumptions are evaluated, which can help feature engineer new columns used for the prediction model and more contextual data. The results are then evaluated manually, which helps provide insight for a simple ML model. The best features picked out are used to predict a single variable. The Random Forest Regression model is trained in 100 iterations and then tested against real data. The model is then trained on the established dataset with the longest continuous period; it is also tested with the shorter time periods found but has around one week of data.

[1] https://www.zabbix.com/integrations/linux

6 Implementation

This chapter aims to explain the setup of the Solution Design in Chapter 4 in practicality. Everything is implemented using Python inside Jupyter Notebook. For the tools used, see Section 2.6.

For the computation of Completeness in Equation (4.2), see Figure 5.

```python
# For counting the completeness of data frames
import pandas as pd

missing_value_per_host = []

for df in host_dataframes:

    # Calculate the total number of entries in the data frame
    total_cells = df.size

    # Count the number of missing values across the entire DataFrame
    missing_values_count = df.isnull().sum().sum()

    # Count the number of zeros in the DataFrame
    zero_values_count = (df == 0).sum().sum()

    # Calculate the total missing value in percentage
    total_missing = (( missing_values_count + zero_values_count ) / total_cells) * 100

    missing_value_per_host.append(total_missing)

print("Total:", sum(missing_value_per_host) / len(missing_value_per_host))
```

Figure 5: This code is used to compute the completeness of each dataset within each host.

For the computation of Timeliness in Equation (4.4), see Figure 6.

```python
# For counting the timeliness in data frame

import pandas as pd
import numpy as np
from datetime import datetime

# Sample DataFrame

data = {
    'timestamp': ['Time of data frame']
}
df = pd.DataFrame(data)
df['timestamp'] = pd.to_datetime(df['timestamp'])

# Define the date of today

today = datetime.now()

# Calculate the age of the data in days
df['data_age'] = (today - df['timestamp']).dt.days

# Define the decline rate
decline_rate = 0.0


# Calculate the timeliness using the given formula
df['timeliness'] = np.exp(-decline_rate * df['data_age'])

# Display the resulting DataFrame
print(df)
```

Figure 6: This code is performed on each dataset for each host to compute timeliness.

The consistency between metrics caused a problem when computing as the metrics are highly varied between the hosts, as seen in Appendix A.1. This meant creating a core set of metrics that a host had to have, and the consistency of metrics was calculated with those in mind. The equation of Consistency (4.3) was modified to the following:

$$\text{Metric Consistency} = \frac{\text{hosts with Metric Collection} \cap \text{Core Metrics}}{\text{Total amount of hosts}} \tag{6.1}$$

For the computation of the modified Consistency of metrics in Equation (6.1), see Figure 7.

```python
    # metrics collected by host / core set of metrics chosen =
    ↪ 1

    def calculate_coverage(comparing_list):
        # Convert lists to sets
        reference_set = set(list_of_core_metrics)
        comparing_set = set(comparing_list)

        # Calculate intersection
        intersection = reference_set.intersection(comparing_set)


        # Calculate coverage proportion
        if not reference_set:
            return 100  # If the reference list is empty, return
            ↪ 100\% coverage by default
        coverage_ratio = len(intersection) / len(reference_set)

        # Convert to percentage
        coverage_percentage = coverage_ratio * 100


        return coverage_percentage

    core_hosts = []
    # for df in host_dataframes:

        # Get the hosts metrics
        metrics = df.columns

        coverage = calculate_coverage(metrics, list_of_core_metrics)
        # Compare it to core set of Metrics

        if coverage == 100:
            core_hosts.append(df['host'])
        else:
            print("Host does not fit the standard!")

    print("Metric consistency:", len(core_hosts) / len(host_dataframes))
```

Figure 7: This code is performed on each dataset to compute the metric consistency for each host.

The time difference in the metric collection was also slightly varied between the hosts. This caused a new equation to be created for time inconsistency, based on the statistical formula for Coefficient of Variation [1]. This tells the story of the general variation in the collection between the hosts. For the computation of the Consistency in time collection, see Figure 8.

[1] https://en.wikipedia.org/wiki/Coefficient_of_variation

```python
# Convert 'clock' to a datetime format if necessary
df['clock'] = pd.to_datetime(df['clock'], unit='s')  # Assuming 'clock' is in seconds since epoch


# Sort by 'hostid' and 'clock' to ensure the order
df = df.sort_values(by=['host', 'clock'])


# Calculate differences in 'clock' for each host
df['time_diff'] = df.groupby('host')['clock'].diff()


# Check variability in time differences for each host
time_diff_variability = df.groupby('host')['time_diff'].nunique()


# Calculate the mean
mean_metrics = np.mean(time_diff_variability)


# Calculate the variance
variance_metrics = np.var(time_diff_variability)


# Calculate the standard deviation
std_dev_metrics = np.sqrt(variance_metrics)


# Calculate coefficient of variation
co_of_var = (std_dev_metrics / mean_metrics) * 100


print(f"Coefficient of Variation of metrics collected: {co_of_var}\%")
```

Figure 8: This code is performed on each dataset to compute the time-varied consistency for each host.

For the application of ACF and PACF, using the Statsmodel and adding the significant lags to the existing datasets, see Figure 9.

```python
    def calculate_acf_pacf_features(series, nlags=10, alpha=0.05):
        acf_values, confint_acf = acf(series, nlags=nlags, alpha=alpha,
        ↪ fft=True)
        pacf_values, confint_pacf = pacf(series, nlags=nlags,
        ↪ alpha=alpha)

        # Filter acf and pacf values based on confidence
        ↪ intervals
        acf_values = np.where((abs(acf_values) >= confint_acf[:, 1]),
        ↪ acf_values, 0)
        pacf_values = np.where((abs(pacf_values) >= confint_pacf[:, 1]),
        ↪ pacf_values, 0)

        return acf_values, pacf_values

    def add_lag_features(df, metric, nlags=10):
        if metric not in df.columns or df[metric].dropna().empty:
            print(f"No data available for metric {metric} after
            ↪ preprocessing.")
            return df  # Return the original DataFrame or handle
            ↪ the case appropriately
        if df[metric].var() == 0:
            print("Variance is zero, which may cause computation
            ↪ issues.")
            return df, None  # or handle as needed

        acf_values, pacf_values = calculate_acf_pacf_features(df[metric],
        ↪ nlags=nlags)
        new_columns = {}
        for i in range(1, nlags + 1):
            new_columns[f'{metric}_acf_lag{i}'] = df[metric].shift(i) *
            ↪ acf_values[i]
            new_columns[f'{metric}_pacf_lag{i}'] = df[metric].shift(i) *
            ↪ pacf_values[i]
        new_columns_df = pd.DataFrame(new_columns, index=df.index)
        df = pd.concat([df, new_columns_df], axis=1)
        df.dropna(inplace=True)  # Drop rows with NaN values
        ↪ resulted from lagging

        return df
```

Figure 9: The code that was used to apply ACF and PACF functions from the Statsmodels module and the addition of the lagged features to the existing datasets.

7 Results

This chapter aims to showcase the results for the *Aims* that were set, to answer the research questions in Section 1.1. The data exploration yielded a few answers before making quality measurements on it. There are two timelines for the data: one before restructuring the hosts to collect more data per time unit and one before that. This split of file structure happened in the summer of 2023; the main difference is that the file sizes are much larger. These results are based on the time before data inflation, as it was seen as more manageable to handle. The constraint of the data is the same, the hardware collected is the same, and the collected metrics are the same. One hour of data after the split held almost as much data as one complete day before the split, which causes issues when seeking to handle connect several files to create a continuous dataset; a con of working from a distance as mentioned earlier in Section 4.1. There could be reasons to look into the latter versions in more detail to see if the results are marginally different. In agreement with Ericsson Research, only the earlier part of the data is explored as the collection remains similar. The timeliness remains the same.

To clarify, no outliers were removed from the data, as during the exploration phase, it became clear that it was not evenly distributed, see Appendix A.3. The aim is to assess the data around the clock. Since the data center is used by people in different time zones, there are no standard business hours to look at directly. However, long periods of inactivity also exist, creating skewed data distribution, as activity depends on when the servers are manned. The metrics are skewed and not prone to being distributed normally. Because of this, many of the outliers detected by normal standard deviation methods also spike activity within the metric data. Because of time limitations to explore these outliers, they are assumed to be part of the workloads and not noise, therefore not removed from the data. The exploration is an aggregated version of the original dataset, as data is collected by the minute and aggregated towards hours and days.

7.1 Aim 1 - Data exploration and preprocessing

Before the preprocessing, the data is assessed and judged in four steps. Accuracy is set to 100% for this sample because it represents real data, meaning every entry is correct according to the definition provided in Equation (4.1). This high accuracy is due to the data being factual and collected from actual machines, ensuring a perfect match between the observed and expected values. Timeliness during the period would adhere to its relevance; since the data structure has not changed, the decline rate is nearly zero or exactly zero, resulting in 100.0% timeliness, indicating perfect relevancy. The data itself is only one year old, and as the machines are said not to have been changed, as well as the method of collecting the data, they are as relevant as the data being collected today. The time inconsistency was to make sure that the machines collected at the same time. The inconsistency is reported to be quite

high based on the coefficient of variation, a well-established method for assessing inconsistency. However, this high value is influenced by a few hosts, as detailed in Appendix A.2. Comparing the intervals at which data points are collected for each host reveals that, over the long term, the inconsistency can indeed be significant. Appendix A.1 shows the metric variance between hosts, and Appendix A.2 details the time variance between hosts. The overall data quality values are presented in Table 1. The hosts vary in the quantity of metrics collected. Consequently, it was necessary to define a core set of metrics primarily guided by the hosts with the most extensive data collection. This involved manually identifying and creating an intersection of these metrics to ensure comprehensive coverage.

Table 1 Data Quality Assessment in %

Completeness	**Metric Consistency**	**Time Inconsistency**	**Timeliness**	**Accuracy**
67.5%	0.15%	87.5%	100.0%	100.0%

Columns that only contained *NaN* were removed, leaving 33 metrics as seen in Table 2. There is also the removal of the hosts that were seen as non-significant. They did not match the criteria for a core set of metrics established; see Table 2, which explains the low value of the Metric Consistency ratio in Table 1.

Table 2 Metric Cleaning of Non-Significant Metrics

Type of Hosts	**Number of Hosts**	**Columns**
Non-significant	34 hosts	2
Significant	6 hosts	33

The aim is to create full metric consistency between the hosts and completeness, as the original dataset had a lot of *NaN* values filling it. Even with the new dataset with fewer hosts, it still had to be checked for completeness, and it lacked it. Creating a continuous dataset needs consistency and completeness, and as seen in Table 3, it contains a lot of time gaps. The results for the total time gap in the dataset before the split are presented in Table 3. A significant portion of this gap can be attributed to downtime during the summer months. This downtime may be due to various factors such as maintenance, reduced operational activity, or seasonal closures.

Table 3 Summary of Gap Data

Number of Gaps	Total Gap Duration (minutes)	Total Gap Duration
115	37499.00	624.98
115	37499.00	624.98
116	37574.00	626.23
115	37609.00	626.82
115	37572.00	626.20
100	37346.00	622.43

The data exploration spurred the finding of a continuous time period with minor time gaps and high data quality. This resulted in finding period with no gaps, creating a time series of 34 days. The data is consistent between the hosts; they all share the same metrics, are complete, and share time consistency. For the established dataset qualities, see Table 4. The total gap duration in the data is omitted; it represents the daily saving of metrics, generating data gaps every day. Time inconsistency is also omitted for these hosts as the recorded data is collected simultaneously. Further findings revealed four weeks of continuous data that met all quality criteria, see Appendix A.8 for metric activity. This uninterrupted dataset is particularly valuable for testing the predictive model, offering a reliable basis. To confirm the occurrence of different activities throughout the week, it is assumed that activity levels are higher during weekdays compared to weekends. A summary of CPU idle time, where lower values indicate higher activity, can be found in Appendix A.4. This analysis motivated the decision to focus on periods with high data integrity further. The 34-day dataset was collected to visualize trends, inspect the data, and highlight the differences between weekdays and weekends. As logically expected, the metrics show more activity during the week than on weekends. For a detailed comparison of how metrics fluctuate across all hosts at two different resolutions—one with a limited Y-axis and one without—refer to Appendix A.5.

Table 4 Established dataset

Hosts	Completeness	Metric consistency	Metrics	Total Gap Duration (Minutes)
6	100.0%	100.0%	33	0.0

Figure 10 illustrates the difference in activity between hosts, specifically by comparing CPU Utilization. CPU utilization is a key performance metric for servers in a data center. It directly measures the computational workload, allowing for the detection of activity on a computer. Due to its importance in monitoring and managing server performance, the inconsistency in measurement can be seen in the scales of the activity. The variety in the data suggests high heterogeneity, further emphasizing the importance of this metric. This result aims to demonstrate that the same metric for different hosts holds a different scale of values.

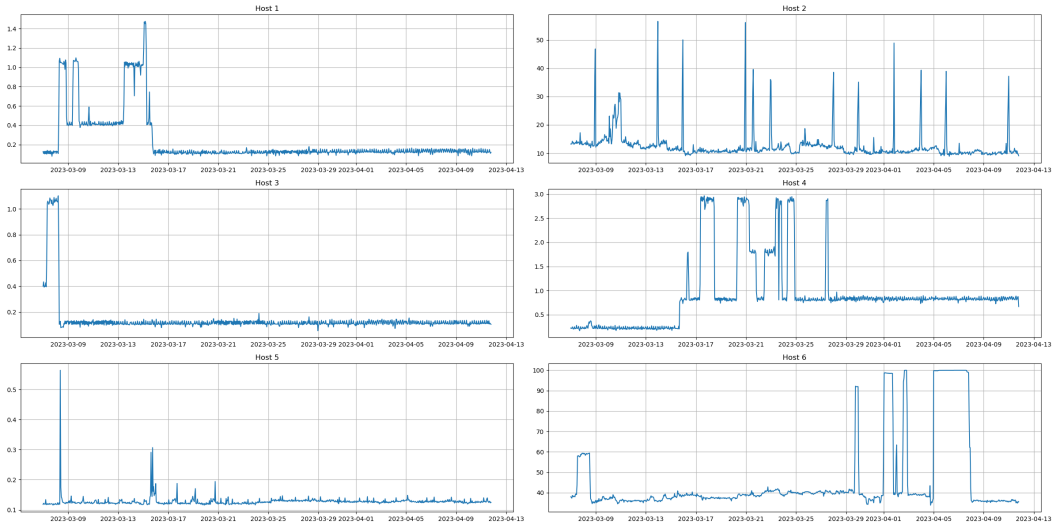


Figure 10: CPU activity shown between different hosts. The Y axis explains the activity for the CPU in percentage, and the X axis spans the time period of the established dataset.

The following figures, Figure 11 and Figure 12 explain the activity of the metrics during the continuous dataset of 34 days. The first week in the plot starts on a Tuesday. Otherwise, each date is weekly, starting on a Monday and ending on a Sunday. It is two hosts' activity, which is chosen as they hold metrics that contain more activity. The plots are limited in scale to show most of the metrics, as there are also units counted in quantity. That means some of the metrics are, i.e., not in percentage, and therefore, the Y axis is scaled much higher than 100, where it counts the frequency of an occurrence. Some spikes can be seen very clearly in Figure 11, which is a clear pattern as they are the inverse of each other. These are the CPU idle time and the metrics for CPU utilization. However, the demonstration of the plots shows that although the week mostly has more activity, spikes occur even during the weekends, where the activity follows the week's trends. Some spikes occur over several days. For full resolutions over the whole time period with no scale limit, see Appendix A.5.

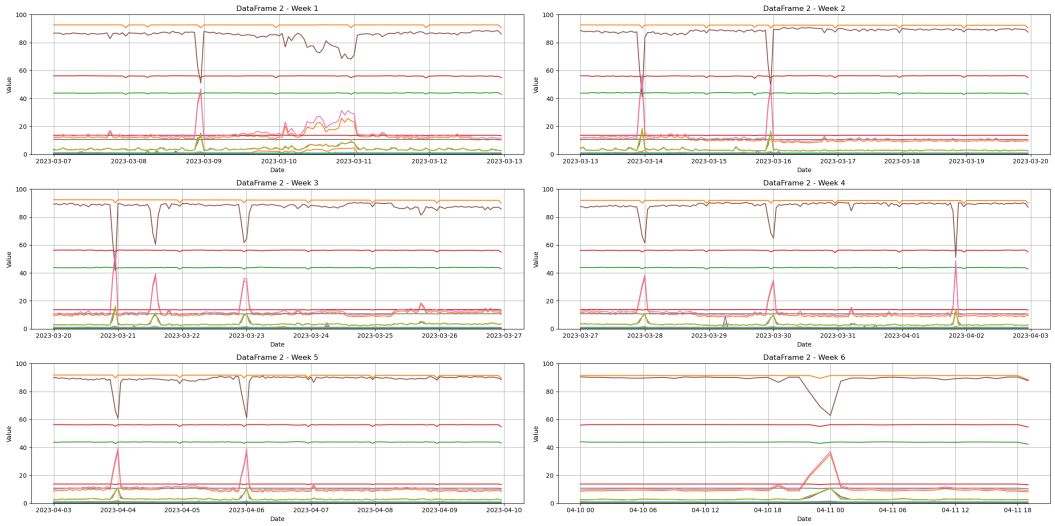


Figure 11: This shows all metric activity under the period of the established dataset. The Y-axis is scaled to 100 to see most metric activities and showcase the value and activity of the metric. The X-axis is the time period. It is based on Host 2.

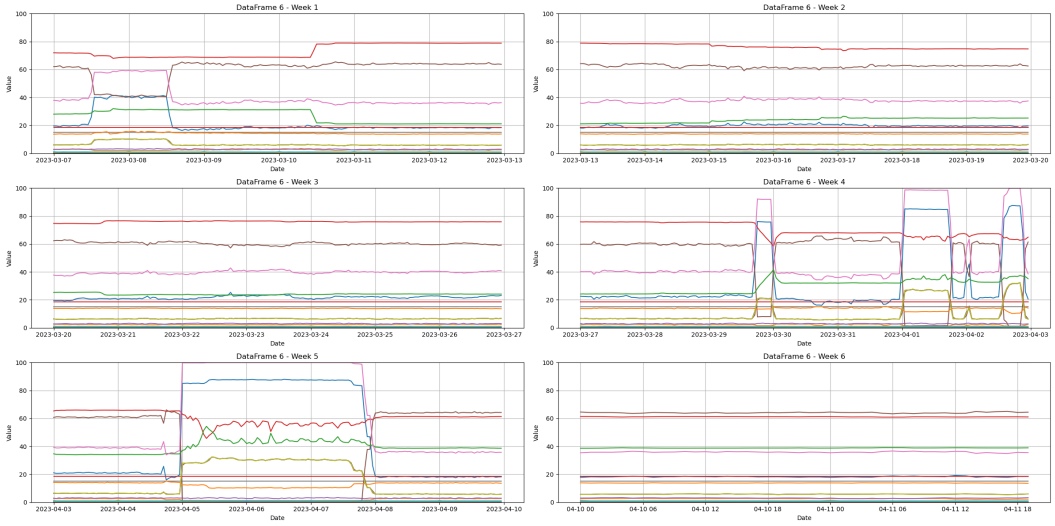


Figure 12: This shows all metric activity under the period of the established dataset. The Y-axis is scaled to 100 to see most metric activities and showcase the value and activity of the metric. The X-axis is the time period. It is based on Host 6.

Looking at Figure 13 and Figure 14, CPU utilization for the same hosts, when some work is started, it can spike and influence the activity in following days.

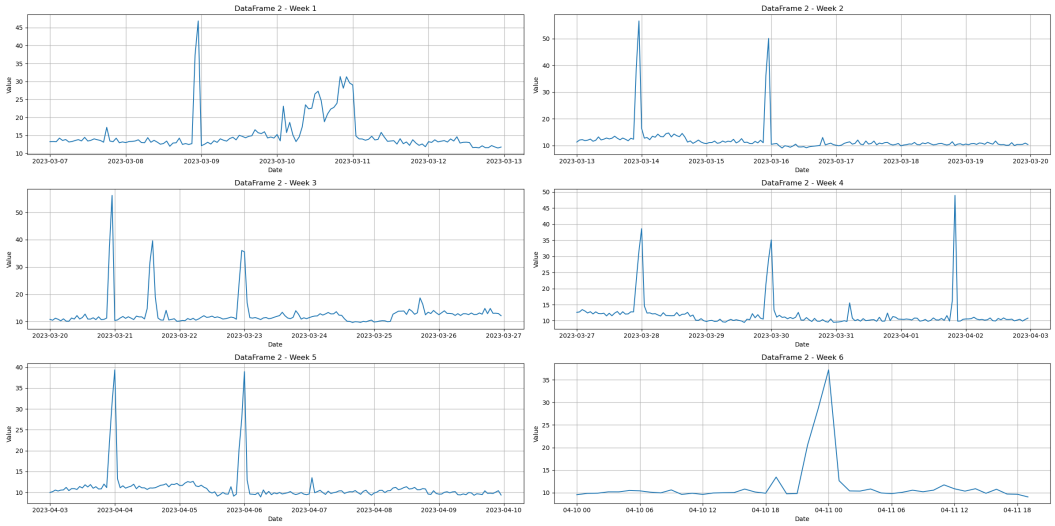


Figure 13: CPU activity shown between different hosts. The Y axis explains the activity for the CPU in percentage, and the X axis spans the time period of the established dataset. It is based on Host 2; it can be seen that once the workload is started, this metric spans over the next day or days.

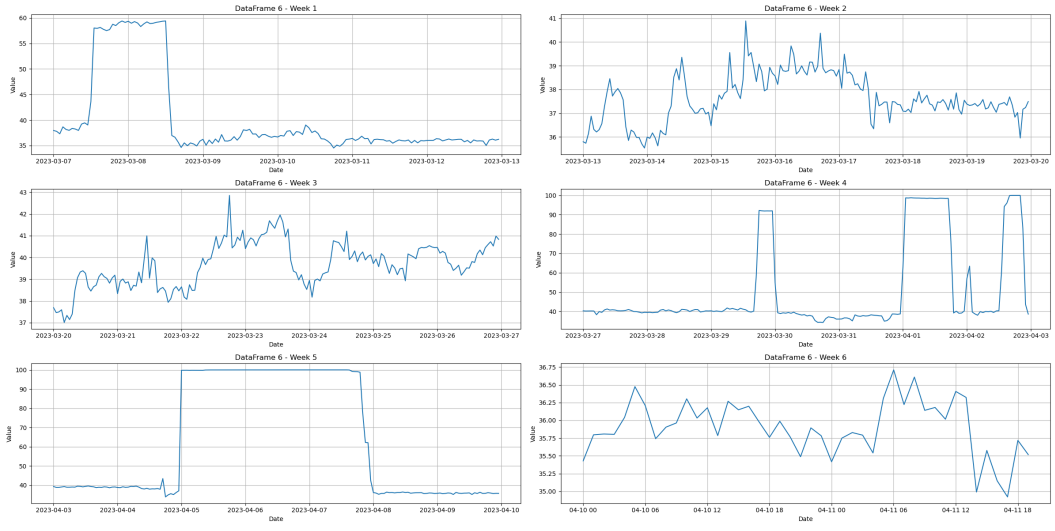


Figure 14: CPU activity shown between different hosts. The Y axis explains the activity for the CPU in percentage, and the X axis spans the time period of the established dataset. It is based on Host 6; it can be seen that once the workload is started, this metric spans over the next day or days.

7.2 Aim 2 - Temporal resolutions and Feature selection

The following graphs discuss the important lags and the more relevant temporal resolution. They summarise the values of using ACF and PACF. The graphs shown summarize all metrics to show relevancy between all hosts. They also summarize all the values in between and how many times they were lagged. The 2D graphs explain the number of times they

were lagged as significantly above a 95% confidence interval. The 3D graphs also explain the values of each lag on average between the times they were counted, that is the correlation and strenght that lag contributes. To see the graphs regarding a daily temporal resolution, See Figure 15 and Figure 16. For the standalone of the 3D graphs, see Appendix A.6.

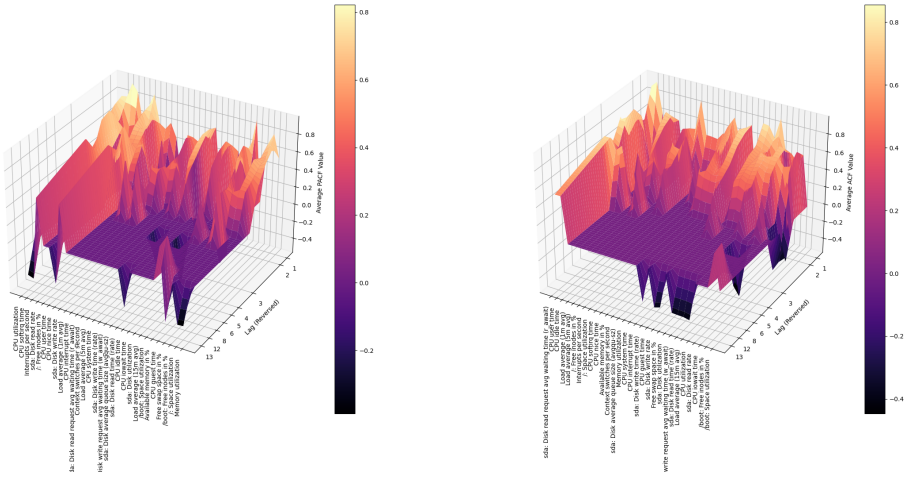


Figure 15: 3D graphs over PACF (left) and ACF (right) significant metrics, where the Y-axis is the value of each lag, the X-axis is the corresponding lag, and the Y-axis is the quantity of the responding metric.

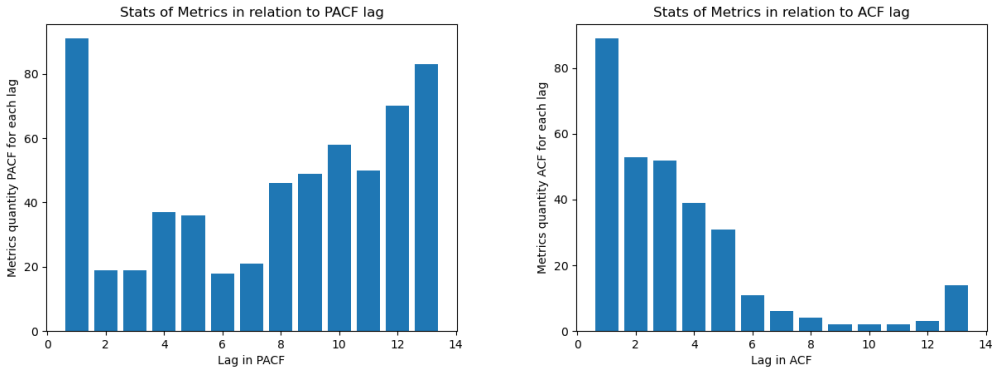


Figure 16: Statistics over PACF (left) and ACF (right) significant metrics, where the Y-axis explains the quantity of metrics measured as significant and the X-axis explains at what corresponding lag.

To see the graphs showcasing an hourly temporal resolution, See Figure 15 and Figure 18. For the isolated plot of the 3D graphs, see Appendix A.7.

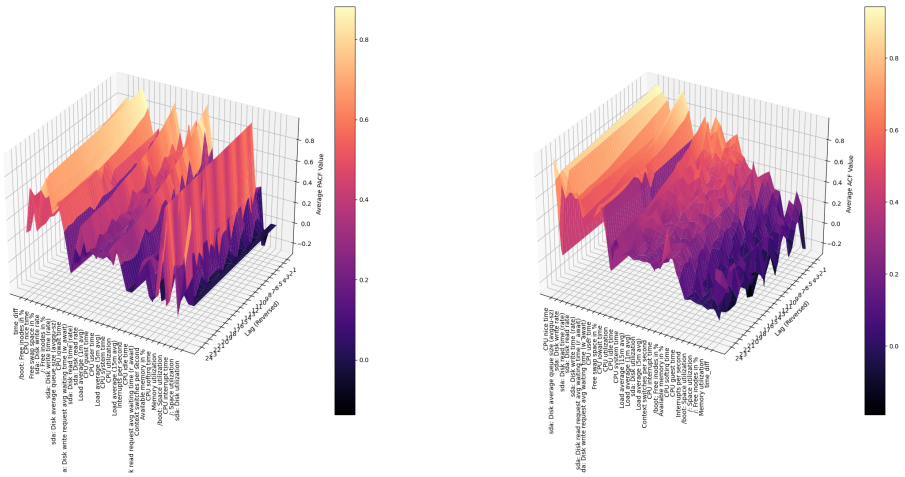


Figure 17: 3D graphs over PACF (left) and ACF (right) significant metrics, where the Y-axis is the value of each lag, the X-axis is the corresponding lag, and the Z-axis is the quantity of the responding metric.

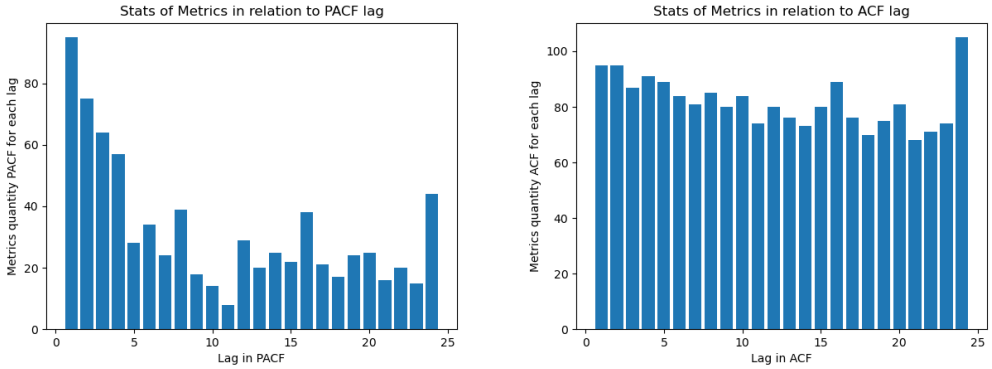


Figure 18: Statistics over PACF (left) and ACF (right) significant metrics, where the Y-axis explains the quantity of metrics measured as significant and the X-axis explains at what corresponding lag.

The following table shows unique metrics seen as significant, that is, over the 95% confidence interval, see Table 5.

Table 5 Significant Unique Metrics found in each host, hourly resolution

Host	Significant PACF metrics	Significant ACF metrics
Host 1	24	24
Host 2	23	24
Host 3	21	21
Host 4	24	24
Host 5	24	24
Host 6	26	26

The following graphs explain the feature selection results and reveal a fingerprint that creates relevant features between all hosts; see Figure 19. The feature selection is only done hourly, as the previous results with ACF and PACF showed more information can be gathered on that temporal resolution. The figure mentioned firstly shows correlations averaged out between all hosts; this means that the values each metric contributes to each unique host are averaged to showcase if there are strongly correlated values between all hosts. Figure 20 showcases a fingerprint of all relevant features between all hosts, that is, without average, and is all layered on top of each other without modifications; this is how the fingerprint would look if the hosts shared metrics that seemed highly correlated.

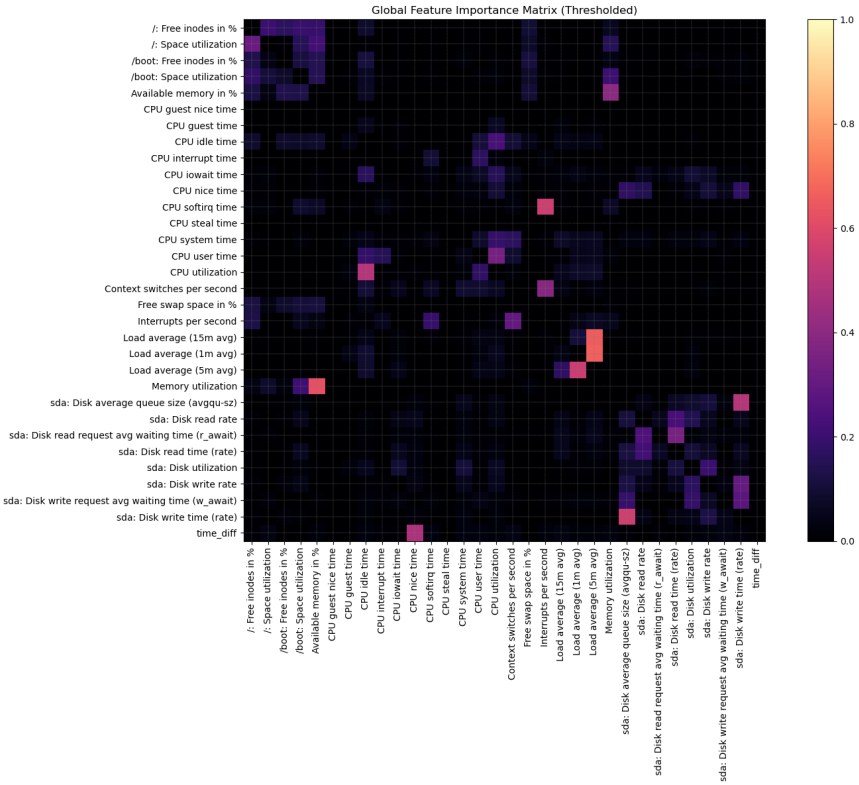


Figure 19: These are the correlated features between all hosts and their corresponding metric data. It is averaged out to be a relative measurement between all hosts. Without a threshold. Not many highly correlated metrics can be seen with all hosts combined.

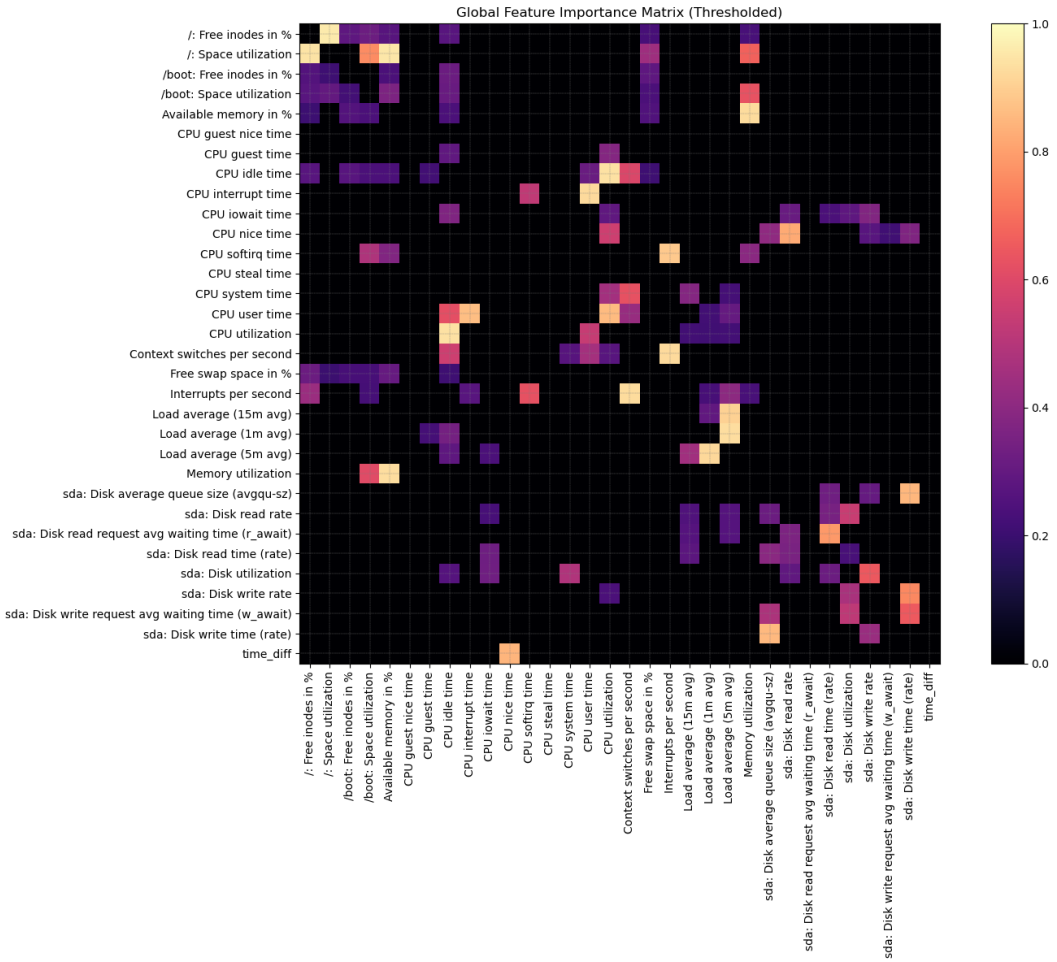


Figure 20: These are the correlated features between all hosts and their metric data. The plot does not have to be average. It showcases the pattern of all features detected as relevant between all hosts, added together. It is thresholded with a value of 0.2.

Furthermore, looking at all the individual feature selections in Figure 21. They are also thresholded with a value of 0.2. The graphs follow the same axis order as Figure 19 and Figure 20.

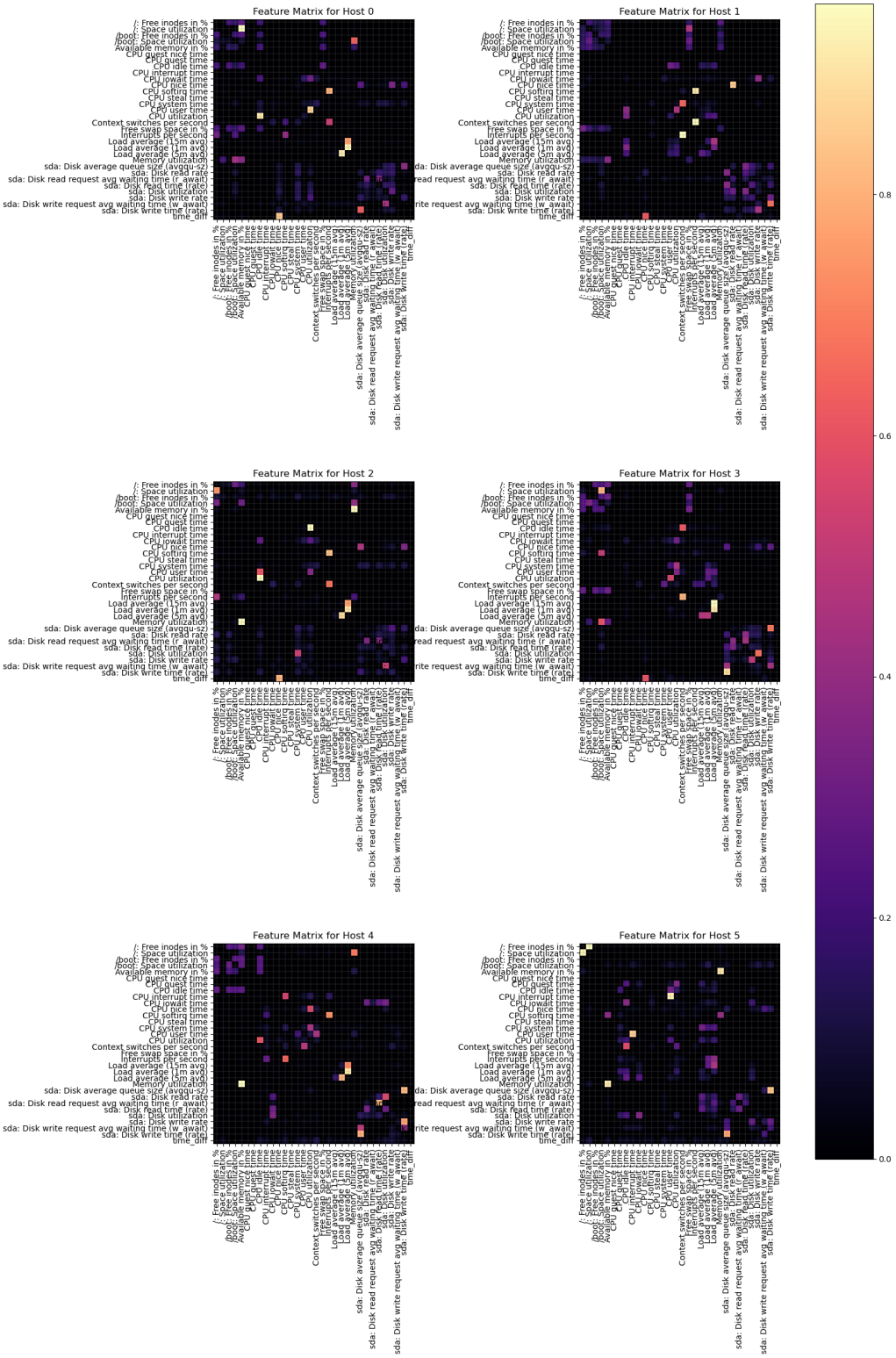


Figure 21: These are the correlated features between all hosts and their metric data. These show all the host's individual correlations towards the metric. It is thresholded with a value of 0.2.

The dataset comprises 33 metrics, as detailed in Table 2. The ML feature selection (F.S) has identified that each metric combination is unique to each host. This uniqueness is a key factor in our analysis. The metrics whose lags were significant have been feature-engineered (F.E) into the dataset with the help of Machine Learning, which has proven to be a reliable tool for predictions. This is a result of combining the machine learning feature selection with the lags seen as significant for each metric with PACF or ACF. The results also change when adding PCA and ACF lag as a feature-engineered metric and the significant lags to the ML model, as shown in Table 6.

Table 6 Feature Selection Stats

Unique F.S Metric per host	F.E ML ACF/PACF per host	Added ACF/PACF metrics
27	37	10
27	34	7
26	34	8
28	34	6
27	35	8
26	54	28

The PCA was performed with 90% variance kept, removing unnecessary noise but keeping most of the data with integrity. The results from the PCA method to analyze the contribution of variance and what metrics might seem more important averaged out over all hosts can be seen in Figure 22. The plot explains what metrics between all hosts contribute to the dataset in variance, which can help give insight into the more important metrics. The PCA plots can be interpreted this way; they create their own pattern in each principal component (PC) on the X-axis. That is how much the metric contributes to the variance of that PC. Since PCA looks at the dataset as a global structure, it can help give insight into what metrics give the most variance, which is often correlated to the importance of that metric. Metrics aligned with each other in each PC can also be correlated if they have a similar variance correlation (not the absolute value). Each PC has its own orthogonal pattern of variance and does not correlate with others. There could be cases where one metric pops up in several PCs, which creates an underlying pattern. They are also thresholded at 0.3, looking at the max value for each scale as it is relevant to that PCA application; the average mean would be 0.6, which means that the set threshold gives room for more metrics.

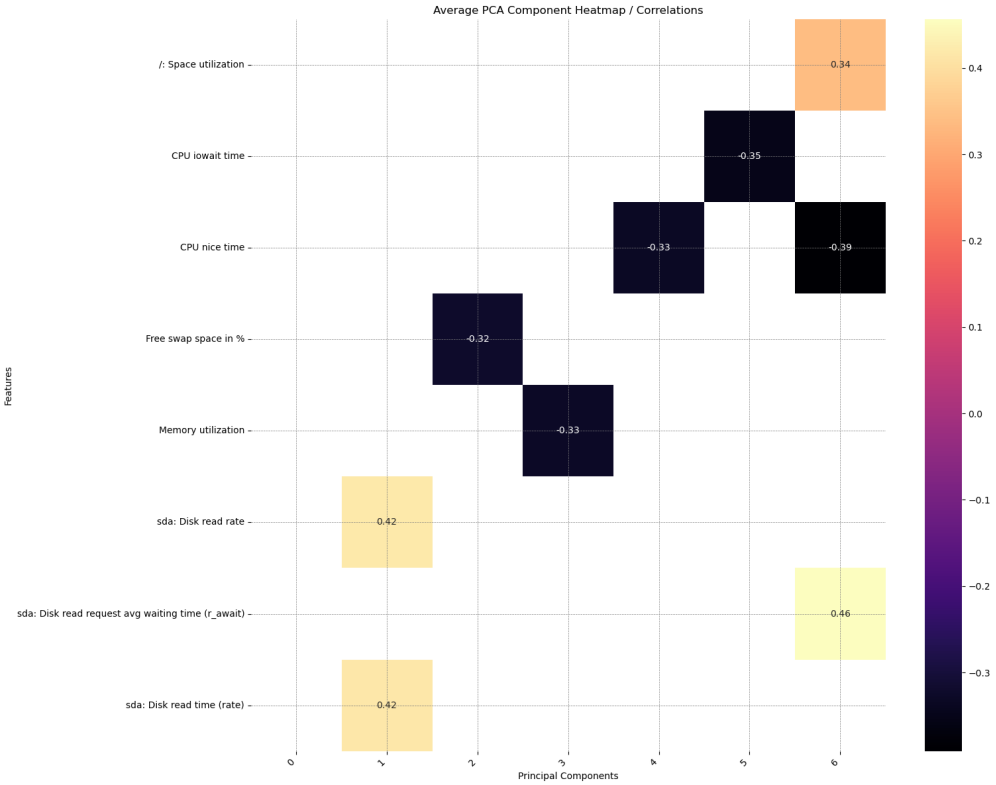


Figure 22: This shows what metrics contribute the most variance between all host datasets, that is, for each principal component created. It is thresholded by 0.3.

In Figure 23 shows how they look independently contributing variance.

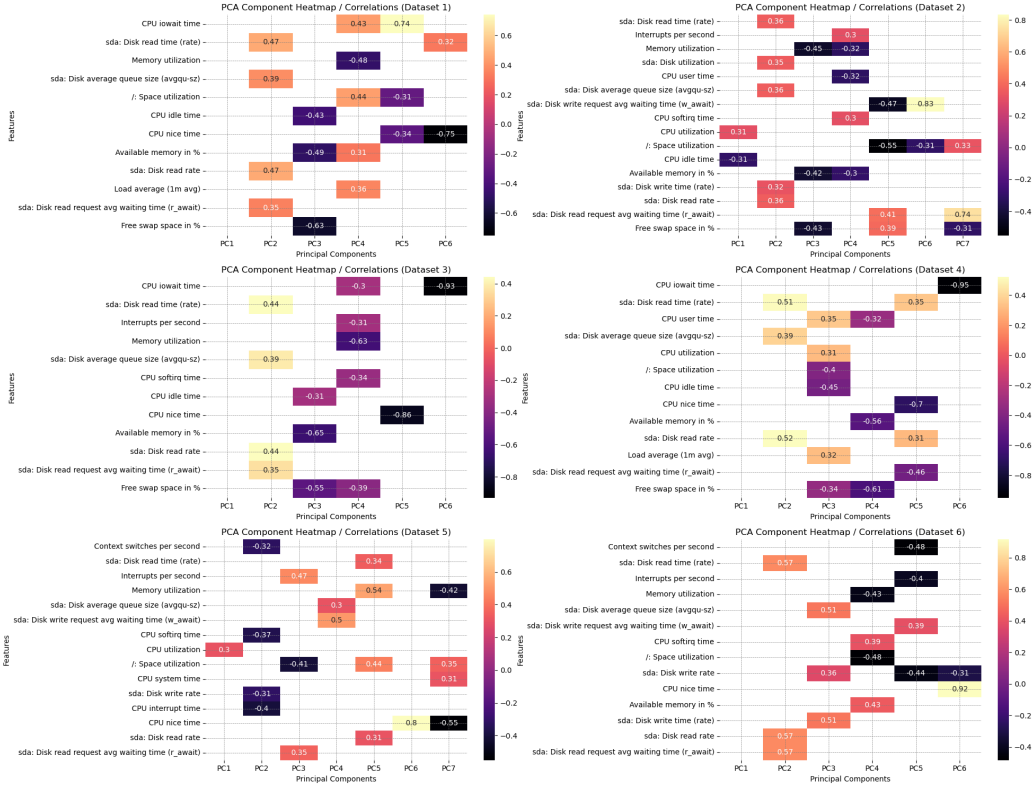


Figure 23: This shows what metrics contribute the most variance between all host datasets, that is, for each principal component created. It is the individual contribution of variance for each unique Host. They share some similarities. It is thresholded by 0.3.

Figure 24 explains the metrics relationships between the different methods applied through the Venn diagram. This shows how they all show different values that might be deemed important to the dataset. The Feature Selection diagram, however, acts differently. As the other methods single out one metric at a time, the feature selection needs at least a pair, inflating the number of metrics and almost always containing a union of all metrics.

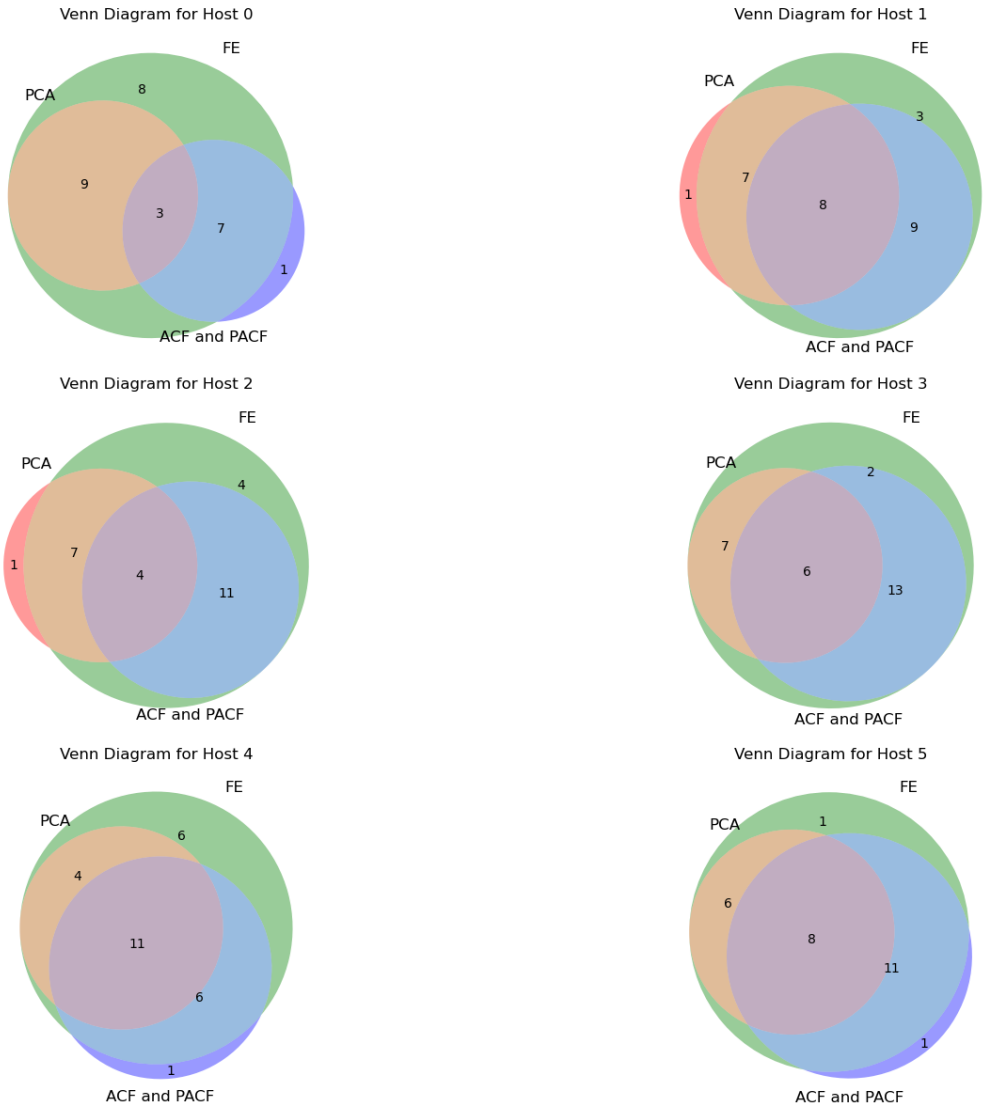


Figure 24: This showcases how many metrics are shared as important between the methods used to explain the importance and individuality between the hosts. The yellow circle is the PCA variance thresholded with a value of 0.3. The green circle is the feature selection thresholded with a value of 0.2. Finally, the blue circle is a union of the metric with significant lags output with ACF and PACF.

7.3 Aim 3 - Machine Learning Model Performance

These are the results from the model; a standard deviation plot showcases the values from both the test data and tested against real data. The prediction experiment is against test and real data; for metric data activity of the real data, see Appendix A.8. However, to have the tests fair, they are all performed on the same host (Host 0). The host has no specific reason for being picked. From earlier figures, it is mostly seen that hosts have high individuality. For the results, all value predictions under zero in predictions are set as 0 in value to keep mean values robust. The figures explain how some metrics are easier to predict than others. All metrics are tested for prediction on the host and the accuracy of their prediction. See Figure 25 for the test data and Figure 26 for the real data.

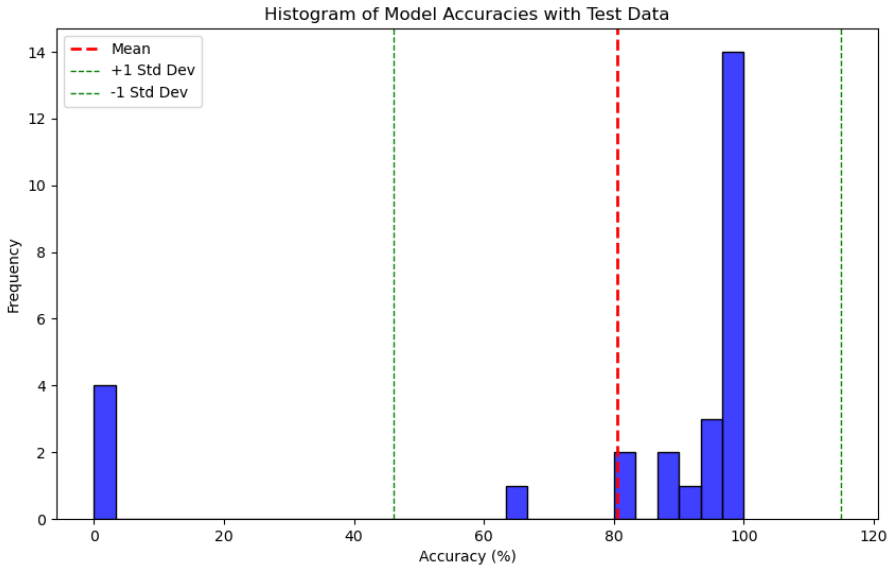


Figure 25: A standard deviation model visually showcasing the accuracy of the predictions made with the model on test data. The Y-axis is used to explain the number of metrics predicted at what accuracy to the X-axis's accuracy.

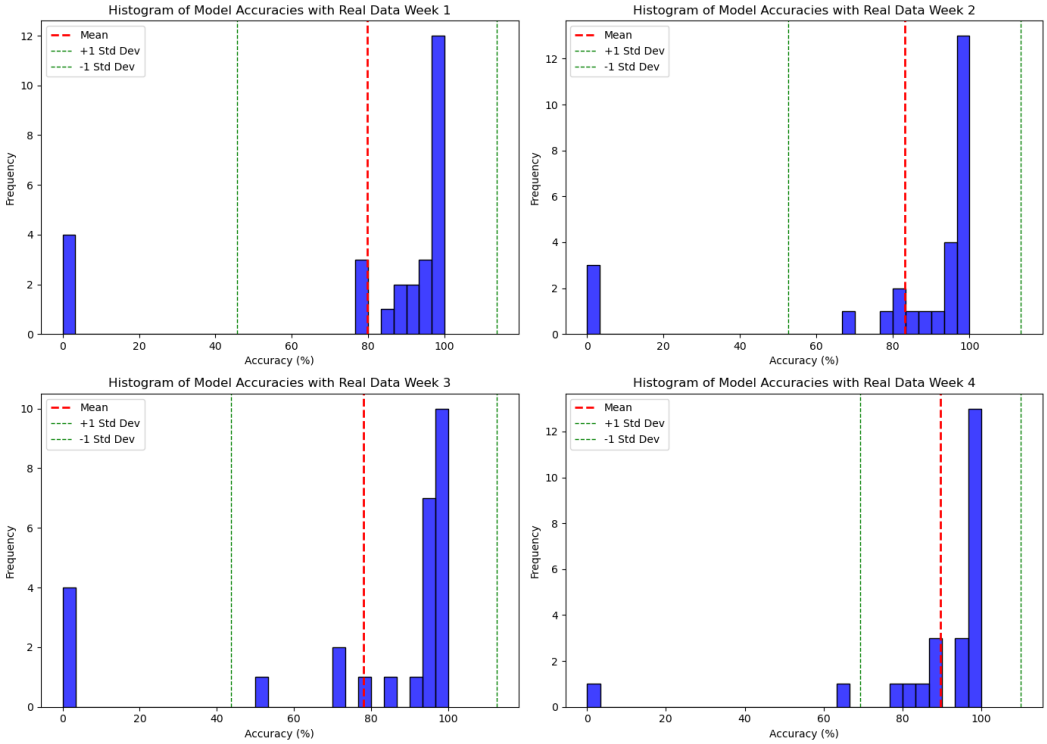


Figure 26: A standard deviation model visually showcasing the accuracy of the predictions made with the model on test data. The Y-axis explains the number of metrics predicted at what accuracy to the X-axis's accuracy. Every plot represents one of the weeks from the real data.

PCA is applied to the same model; PCA helps reduce noise after all features are selected, that is, to reduce dimensionality when variables are already chosen for prediction. The PCA is applied with the parameter of keeping 90% of the variance to hold the important information intact. The results are different; See Figure 27 for test data results and Figure 28 for the real data. All results under zero in predictions are set as 0 in value.

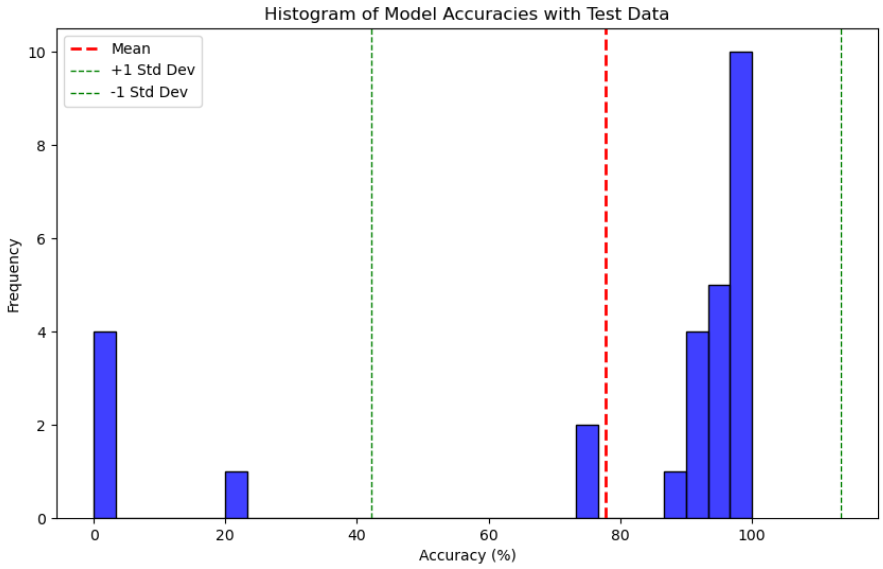


Figure 27: A standard deviation model visually showcasing the accuracy of the predictions made with the model on test data. The Y-axis is used to explain the number of metrics predicted at what accuracy to the X-axis's accuracy. This is with PCA applied to the dataset.

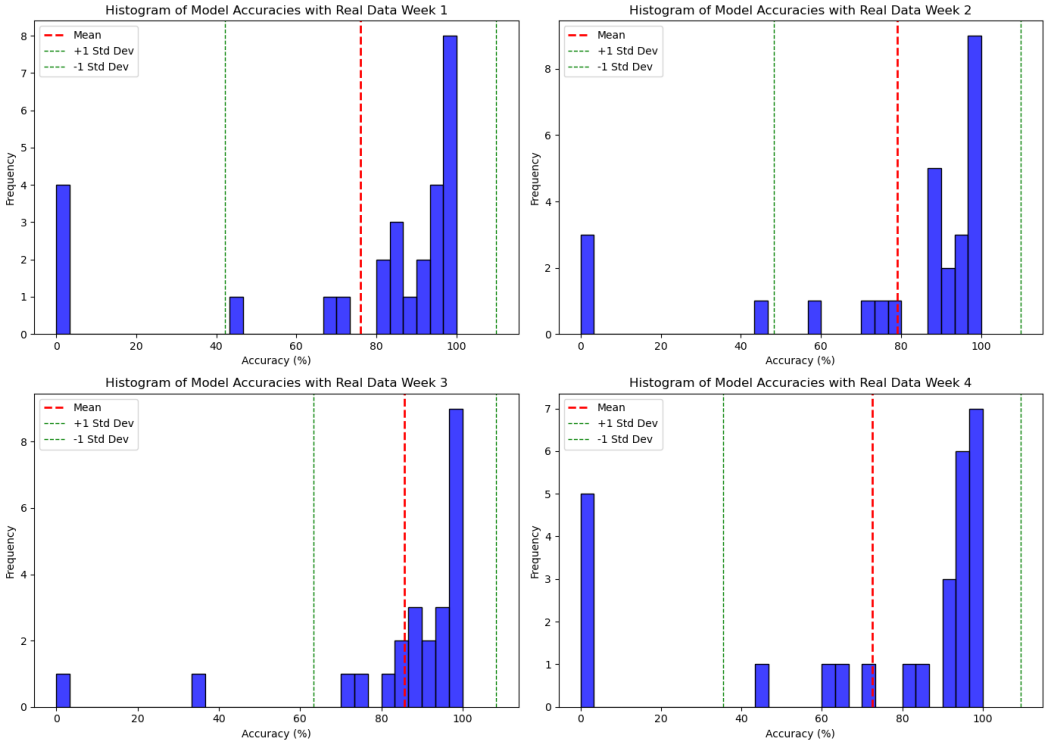


Figure 28: A standard deviation model visually showcasing the accuracy of the predictions made with the model on test data. The Y-axis explains the number of metrics predicted at what accuracy to the X-axis's accuracy. Every plot represents one of the weeks from the real data. This is with PCA applied to the dataset.

8 Discussion

The innovative nature of this study lies in its practical application of predictive analytics to a real-world dataset from Ericsson Research. Unlike theoretical research, this case study demonstrates the tangible benefits and challenges of using Zabbix-collected metric data in an industrial setting. The methodologies employed, such as feature selection and temporal resolution analysis, are tailored to address the specific needs and constraints of the company, providing actionable insights that can be directly implemented to improve operational efficiency.

Starting from the first point, this study addresses the research questions in Section 1.1, by formulating specific Aims and objectives to navigate the broad field of metric data analysis. The first research question—"What metrics from the data center can be used for forecasting, and which metrics hold value and relevance?"—is addressed through all three aims. Aim 1 seeks to filter out unnecessary metrics and retain those of quality. Aim 2 evaluates these metrics' predictive suitability and behavior, determining their value for forecasting. Aim 3 examines the practical application of these findings and explores additional insights. The second research question—"Which specific metrics are most valuable for predictive analysis from the collected data?"—is tackled by using Aim 1 to eliminate preliminarily useless metrics and ensure data quality, Aim 2 to assess useful metrics for feature selection and relevance, and Aim 3 to test the practicality of these metrics with a simple model. By structuring the aims as a guideline, the study discusses the results of each aim in detail, ultimately answering the research questions rather than merely fulfilling the aims

Moving on to the first results, the aim is to assess the data quality and answer what metrics are redundant. In Table 1, the time inconsistency was given too much focus. As it varies to some degree between hosts, it is still collected at a similar interval at the machine's local time. The completeness and metric consistency in Table 1, was not expected. The time gaps are consistent between hosts, which is assumed to be a general downtime in the data center, although this could not be confirmed. To have as much integrity as possible to the data without performing imputations or artificially manipulating the data, this gave a much lower continuous dataset than assumed. It is 34 days of continuous data without downtime, other than the consistent time gap at the end of each day, assumed to be for some data cleanup, but again, not confirmable, which in turn created the established dataset seen in Table 4.

Interesting patterns were also seen during the data exploration. CPU utilization is often a metric that shows activity performed on a computer, as modern computers all use the CPU to perform any activity or calculation. In Figure 10, it can be seen that the metrics vary a lot; now, this is not the same for every metric, as some had too little variety and predictable patterns, which make them less interesting. The results show that the spikes are not as cyclic on a first look or have obvious seasonality. The spikes are sporadic, mostly host-dependent, and even for CPU utilization metrics; some have very low variance and seem more static. This would presume high individuality and no general assumption can be made about the

general metrics for all hosts.

It was also assumed that the weeks should have more activity or the metrics should be higher in value. This is not always the case; generally, there seems to be more activity, but they do not always differ much if looking at week 4 in Figure 12. However, on exploring this case, a lot of metric activity seems to spread into the next day, even onto the weekends, i.e., if work is performed on a Friday, it continues on Saturday, which is seen in both Figure 12 and Figure 14. This could be because of the different time zones or processes left running for a while. Looking at it from a higher resolution, it could be assumed that sometimes processes are kept running longer, as seen in Figure 10. Now, this does fulfill the criteria that metrics from the exploration phase hold variety and are quite individual; a continuous time-series dataset qualifies for further exploring, which fulfills success for Aim 1, see Section 4.2. It also clears up the Research Questions in Section 1.1; there are metrics with continuous history and variety. This means they do not always act the same and have a reason to be predicted.

The second part of the results is for determining the importance of some features and the strength they hold for predictions, which is by using ML feature selection and the statistical tools ACF and PACF, see Section 4.3. Firstly, the results from the daily resolutions in Figure 16, a third dimension added to Figure 15. On exploration of the metrics, some have more predictable patterns, as mentioned earlier; however, as this is still a high-level exploration, assuming to remove them as of that moment seems too presumptuous. The graphs indicate fewer significant lags than Figure 18. This suggests a greater benefit of predicting on an hourly basis than a daily one. However, there is also clearly a falloff in the quantity for ACF in Figure 16, which suggests that the intermediate values between the lags do not strengthen the correlation as time passes. The PACF value seems more significant, which could suggest a bi-weekly trend. As stated earlier, metrics follow the usage of earlier days, which could mean on a new week, the workflow is changed, which creates new patterns. Referring to the 3D plot, Figure 15, it can also be seen that the 2D plots can be misleading, as the values (correlation strength) fall off for most metrics, the further the lags go. In most cases, it can be seen that on a daily resolution, most metrics fall off after the third lag, which would follow the manual inspections on most metrics that seemingly have a 2 or 3-day trend of stronger correlation before falling off. The key metrics contributing to the underlying patterns remain consistent across daily and hourly resolutions. Disk I/O, memory utilization, and CPU metrics are crucial at both temporal granularities.

The difference between the daily and hourly resolution metrics is that significantly more lags are detected hourly; however, the correlations' strength is also more detectable. However, the following trend is that most metrics hold strong historical data on a 24-hour basis. As seen in PACF, the direct correlations seem weaker, which suggests that the intermediate values hold strong values to create strong correlations for the next hours. Looking further into how many unique metrics are detected on each host, see Table 5; there is a similarity in the number of metrics suggested to be strongly correlated in the time series. The comparison shows that hourly resolution data provides stronger and more immediate historical values detectable through ACF and PACF analysis, making it more effective for short-term monitoring and rapid response. While useful for long-term trend analysis, daily resolution data may not capture the finer, immediate fluctuations as effectively as hourly data. The lags for daily resolution reveal strong immediate dependencies and could indicate weekly patterns; however, it suggests a weekly cycle or reset. For real applications, combining both

resolutions would yield the most robust approach to performance management, balancing real-time responsiveness with long-term planning. This study, however, moved to focus on the hourly resolution as it held more data.

Moving on to the stats and fingerprints, they are performed on an hourly basis, as seen with ACF and PACF; there is more historical data there, and when aggregating it to a daily resolution, information can disappear. Another thing that suggests that it is hard to draw a general comparison of what metrics are more important is Figure 19. The fingerprint is all metrics correlations averaged out between them, which suggests about five highly relevant metrics between hosts. Looking closely at what metrics are correlated to each other, three of them are redundant as they are variates of their own metric; all Load Averages are part of CPU Utilization. However, aggregating the CPU utilization to a lower resolution makes it relevant for prediction. On inspection of the heatmaps individually in Figure 21, it could also be seen that some hosts share some features while others do not. This means that when averaging out the values, the machines that hold similar values get flushed out by the number of other hosts that do not hold that metric value.

Table 6 explains the number of unique metrics picked out per host to be valuable. That means that out of the 33 metrics, a maximum of 26 metrics were seen as important, which would reduce the number of metrics needed for predictions with context. However, putting it into context, if a metric were significant at every lag hourly, it would generate 24 extra features per metric, which in total for all metrics would be 624 extra features (assuming it is also the host with max features, 26). For that host, however, only 9 extra features were added. This is about $1,5\%$ of the capability of the best-case scenario. Looking back at the research question, Section 1.1, it was hard, even with the statistical analysis, to draw a complete conclusion from the results. The metrics act highly individual for the hosts, but it still suggests that some metrics are more important and some hold stronger historical data. The higher temporal resolution holds more predictability. This falls into the criteria for success to be defined for Aim 2 in Section 4.3, where patterns, temporal resolution, and hope of historical data were supposed to be found. In summary, disk I/O and memory utilization are the most critical factors between the host datasets, with CPU metrics also playing a significant role.

The same case is for the PCA average heatmap in Figure 22 as the FE heatmap seen in Figure 19, the metrics become flushed out between the datasets. However, some metrics stand out more in their variety between all datasets. This dominant pattern could be assumed to strongly influence the general pattern of the datasets for all hosts. This is the first result given that is more conclusive between all hosts. However, one must remember that each PC is an orthogonal pattern not connected to other PCs, which means that for the second PC, two metrics pop up with a high similarity. They are also shown to be connected to the disk, which creates a pattern. This is the PC with the second-highest variance in all the datasets. An important finding is that no metrics can be seen in the first PC, the component with the highest variance. Looking at the individual plots in Figure 23, only two hosts contain metrics that correlate with the threshold of 0.3. This could mean that the datasets are too constrained, but essentially, no metric in the PC that holds the most variance seems to influence the most among the datasets. The PCA variance analysis underscores the critical role of disk I/O and memory utilization in influencing system performance. CPU metrics also play a significant role but contain more variability. Combining the insights from PCA and feature selection analyses provides a holistic view of the key metrics between hosts. Disk I/O and memory utilization are the most critical factors for monitoring and optimizing

system performance. CPU metrics also play a significant role but require a more tailored approach.

The Venn diagrams in Figure 24, highlight key insights into system performance metrics across different hosts and analytical methods. Several metrics are consistently identified by all three methods, underscoring their reliability as critical performance indicators. Each method also identifies unique metrics, demonstrating the value of a multi-faceted approach to capture a comprehensive set of influential factors. The variability in unique and common metrics across hosts suggests that specific configurations and workloads affect which metrics are most important, necessitating tailored applications between hosts. Prioritizing metrics consistently identified as important across all methods and hosts can yield significant insights into the underlying pattern. Thus, leveraging the combined strengths of ACF/PACF, PCA, and Feature Selection provides a robust framework for identifying and optimizing key system performance drivers, ensuring comprehensive monitoring and tailored optimization for each host. Clearly connecting to the research question in Section 1.1, when wanting to answer what specific metrics are more valuable.

For the last *Aim* with the knowledge and assessment of the previous patterns and metrics features that seemed strong. An attempt at a simple baseline regression model is made. No heavy modifications are done to the datasets other than feature engineering significant lags as they can help the model with more accurate predictions. The previously established dataset is the one being trained, where the results for the test dataset can be seen in Figure 25. Now, the mean for the results is not in line with any results from the test, where the majority of the metrics range from 96% to 99% accuracy, which is usually a sign of overfitting in the model. A similar pattern can be seen with the results on real data in Figure 26. The plot is larger because it is tested on four individual weeks of data. These are not the results one would hope to get, as they are too optimistic, with a few metrics seemingly unpredictable. However, the frequency of high predictability is concerning and is certainly a result of overfitting. Another reason for these results could be that some metrics, such as CPU Utilization, can be very strongly correlated, such as Idle Time, see Figure 20. They are, per definition, almost the opposite; when one lowers, the other raises; by not removing this feature to predict the other variable, one could also assume the results would be highly accurate. Some metrics are more stagnant than others; that is, they perform similarly over longer periods and do not seem to change their patterns so much, which would contribute to being a metric that is easy to predict. With these results, one would remove these features from the difference in results for further testing. However, to connect this to the criteria for success, see Section 4.4, which is useful information. It was possible to perform predictions. However, the results might not have been as satisfying as hoped, and the model is most likely overfitted.

However, applying PCA to this is more reasonable; assumingly, it removes much of the noise by reducing the dimensionality. It can be seen that the values are more spread out and not always as accurate, which would be more realistic. It could be better with more domain expertise in ML and reasonable accuracy. Future research could apply more cross-validations, as these are simple models only evaluated by accuracy with MSE (Mean Square Error). A summary of all real results, whether applied with PCA or not, seems to be a wider division of accurate predictions, where some metrics clearly perform worse when predicting with real data. It can also be seen that week three generally performs better. Looking at the activity for the metrics there, it is seen that they are very stagnant. This is probably an easier prediction and, hence, gives clearer results.

9 Conclusion

This thesis explored the potential of using computer performance metrics for predictive analysis within a cloud data center, focusing on data collected by Zabbix in the Ericsson Research data center. Through a structured approach involving data quality assessment, feature selection, and model development, the study aimed to answer key research questions about the utility and value of these metrics.

Aim 1: Assess the quality and predictive suitability of Zabbix-collected metric data.

By addressing the first research question—"What metrics from the data center can be used for forecasting, and which metrics hold value and relevance?"—the study focused on filtering out unnecessary metrics and retaining those of high quality. The quality assessment revealed that a core set of metrics with high completeness and relevance could be identified despite inconsistencies and gaps. This ensured that the data used for further analysis was robust, fulfilling Aim 1 and confirming that valuable and relevant metrics exist within the dataset.

Aim 2: Assess predictive capabilities with well-established industry analytical techniques for predictive feasibility.

To answer the second research question—"Which specific metrics are most valuable for predictive analysis from the collected data?"—the study employed techniques such as ACF, PACF, and PCA to evaluate the predictive strength of various metrics. The analysis demonstrated that certain metrics exhibited strong temporal patterns and correlations, indicating their suitability for predictive modeling. The individuality of the hosts suggested that models might need to be tailored to specific hosts to achieve optimal performance. The exploration found that disk I/O, memory utilization, and CPU metrics consistently played a significant role. This exploration fulfilled Aim 2, identifying the most valuable metrics for predictive analysis.

Aim 3: Develop and validate an inference model using identified metrics and methods.

The third aim involved developing and validating a baseline inference model to test the practical application of Aims 1 and 2 findings. By incorporating the most predictive features and temporal patterns identified, the model aimed to forecast future trends. The initial results showed promising accuracy but also indicated potential overfitting. The application of PCA helped reduce noise and dimensionality, resulting in more realistic accuracy levels. This addressed the research questions by demonstrating the practical feasibility of using selected metrics for prediction, fulfilling Aim 3.

This thesis demonstrates the potential of using performance metrics for predictive analysis

in a cloud data center. The findings provide a foundation for future research and development, suggesting that predictive capabilities and operational decision-making in industrial settings can be enhanced with further refinement and more sophisticated modeling techniques.

10 Future Work

This thesis has opened several avenues for further investigation into the predictive capabilities of IT operations metrics. The exploration highlighted the necessity of having a complete and continuous dataset for effective forecasting. However, gaps in data completeness have posed significant challenges. Future studies could explore various imputation techniques or synthetic data creation to address these gaps. Enhancing the dataset's completeness could improve model training and provide a richer, more nuanced understanding of the data.

Many of the metrics analyzed exhibited relatively static values, suggesting potential utility in anomaly detection applications. These metrics could serve as indicators for system health monitoring. Further research should focus on validating and refining the identification of outliers, ensuring that the metrics used can reliably signal deviations that are truly anomalous and worthy of attention.

The global nature of the data center's operations, spanning multiple time zones, introduces additional complexity and opportunity. Further studies could analyze the impact of geographical and temporal factors on data metrics. Understanding the workflow and peak activity times of various teams worldwide could provide insights into operational demands and help optimize resource allocation across different regions.

Additionally, the variability in file types before and after the dataset's summer split offers a unique opportunity to study the effects of data structure changes on the analysis. These differences could reveal how data capture and categorization changes impact the predictive models' performance. This exploration might also uncover more about the interrelationships between different types of data and how they inform the behavior of various hosts within the network.

Moreover, the second split of the dataset merits specific attention to determine if it demonstrates greater variance or enhanced predictive capabilities compared to the first. This comparative analysis could help identify which data characteristics are most beneficial for predictive modeling and lead to more robust forecasting techniques.

Future work could involve more detailed cross-validation, the application of additional machine learning models, and a deeper investigation into the specific patterns and behaviors of different metrics across various hosts. This would help to build more robust and generalizable predictive models, ultimately improving the efficiency and reliability of IT operations.

In conclusion, the breadth of potential research stemming from this initial exploration is vast. Each area offers a path toward deeper understanding and more effective use of IT operational metrics in predictive analytics. By continuing to build on the foundational work laid out in this thesis, future research can advance the findings, driving toward more proactive and informed IT operations management.

10.1 Preliminary Exploration

In data exploration, it is common to use clustering as an unsupervised machine-learning method to identify natural groupings. This technique was employed in the early phases of the project. However, considerable domain knowledge, instinct, and experience are necessary for effective clustering. Clustering relies on the natural grouping of data points based on their similarities. Some iterations produce different clusters for similar data points, which is harder to interpret. This prompted the search for more standard statistical tools that provide more definitive answers. This could mean that applying more domain expertise could be a valuable way to look at the data again.

References

[1] J. Andrews and P. McNicholas. Variable selection for clustering and classification. *Journal of Classification*, 31:136–153, 2013.

[2] L. Ehrlinger, V. Haunschmid, D. Palazzini, and C. Lettner. A daql to monitor the quality of machine data. In *Proceedings of the International Conference on Database and Expert Systems Applications (DEXA)*, volume 11706 of *Lecture Notes in Computer Science*, pages 227–237, Cham, 2019. Springer.

[3] Lisa Ehrlinger and Wolfram Wöß. A survey of data quality measurement and monitoring tools. *Frontiers in Big Data*, 5, 2022.

[4] Christina Ellis. Number of trees in random forests. `https://crunchingthedata.com/number-of-trees-in-random-forests/`, 2022. Accessed: 2024-05-20.

[5] Arash Erfani, Tohid Jafarinejad, S. Roels, and D. Saelens. Linking dataset quality and mpc in buildings: impact of temporal resolution. *Journal of Physics: Conference Series*, 2654, 2023.

[6] J. H. F. Flores, P. Engel, and R. Pinto. Autocorrelation and partial autocorrelation functions to improve neural networks models on univariate time series forecasting. *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2012.

[7] Asst Hamdard and Hedayatullah Lodin. Effect of feature selection on the accuracy of machine learning model. *INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH AND ANALYSIS*, 06, 09 2023.

[8] John Hauser and Gerry Katz. Metrics: You are what you measure! *European Management Journal*, 16:517–528, 10 1998.

[9] Holger Hinrichs. Datenqualitätsmanagement in data warehouse-umgebungen. pages 187–206, 01 2001.

[10] J. Hou, Na Wang, Kaihua Guo, Donglai Li, H. Jing, Tian Wang, and R. Hinkelmann. Effects of the temporal resolution of storm data on numerical simulations of urban flood inundation. *Journal of Hydrology*, 589:125100, 2020.

[11] Liangyuan Hu, Jung-Yi Joyce Lin, and Jiayi Ji. Variable selection with missing data in both covariates and outcomes: Imputation and machine learning. *Statistical Methods in Medical Research*, 30:2651 – 2671, 2021.

[12] IBM. What is principal component analysis (pca)? `https://www.ibm.com/topics/principal-component-analysis`, 2023. Accessed: 2024-04-23.

[13] IBM. Big data analytics. https://www.ibm.com/analytics/big-data-analytics, n.d. Accessed: 2024-02-25.

[14] IBM. Supervised vs. unsupervised learning. https://www.ibm.com/blog/supervised-vs-unsupervised-learning/, n.d. Accessed: 2024-04-30.

[15] IBM. What is a data center? https://www.ibm.com/topics/data-centers, n.d. Accessed: 2023-12-29.

[16] IBM. What is ai? https://www.ibm.com/topics/artificial-intelligence, n.d. Accessed: 2024-02-24.

[17] IBM. What is aiops? https://www.ibm.com/topics/aiops, n.d. Accessed: 2024-02-26.

[18] IBM. What is machine learning? https://www.ibm.com/topics/machine-learning, n.d. Accessed: 2024-02-26.

[19] IBM. What is random forest? https://www.ibm.com/topics/random-forest, n.d. Accessed: 2024-04-30.

[20] ISO/IEC. Software engineering - software product quality requirements and evaluation (square) - data quality model. https://www.iso.org/standard/35736.html, 2008. ISO/IEC 25012:2008, Accessed: 2024-04-30.

[21] J. Jaiswal and Rita Samikannu. Application of random forest algorithm on feature subset selection and classification and regression. *2017 World Congress on Computing and Communication Technologies (WCCCT)*, pages 65–68, 2017.

[22] A. C. Jishag, A. P. Athira, Muchintala Shailaja, and S. Thara. Predicting the stock market behavior using historic data analysis and news sentiment analysis in r. In Ashish Kumar Luhach, Janos Arpad Kosa, Ramesh Chandra Poonia, Xiao-Zhi Gao, and Dharm Singh, editors, *First International Conference on Sustainable Technologies for Computational Intelligence*, pages 717–728, Singapore, 2020. Springer Singapore.

[23] L. Kabari. Principal component analysis (pca) - an effective tool in machine learning. 9:56–59, 2019.

[24] Fasih Khan. Data exploration: A comprehensive guide. https://www.astera.com/type/blog/data-exploration/, 2024. Accessed: 2024-05-03.

[25] Jake Lever, M. Krzywinski, and Naomi Altman. Points of significance: Principal component analysis. *Nature Methods*, 14:641–642, 2017.

[26] Matplotlib. Matplotlib – visualization with python. https://matplotlib.org/, 2024. Accessed: 2024-04-22.

[27] Grzegorz Mentel and Jacek BroÅ¼yna. Historical data in the context of risk prediction. *International journal of business and social research*, 4:48–60, 2014.

[28] Bhavik R. Bakshi Mohamed N. Nounou. Autocorrelation function. Data Handling in Science and Technology, 2000. Accessed: 2024-04.15.

[29] Chris Moran. What makes a good metric? the golden rules of measuring what matters. Medium, August 2018. Available at: https://medium.com/@chrismoranuk/golden-rules-what-makes-a-good-metric-a96045d7ab24, Accessed: 2024-04-23.

[30] Oracle. What is big data? https://www.oracle.com/big-data/what-is-big-data/, n.d. Accessed: 2024-02-24.

[31] T. Oshiro, P. S. Perez, and J. A. Baranauskas. How many trees in a random forest? pages 154–168, 2012.

[32] pandas. pandas - python data analysis library. https://pandas.pydata.org/, 2024. Accessed: 2024-04-22.

[33] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[34] Penn State University. 2.2 partial autocorrelation function (pacf). https://online.stat.psu.edu/stat510/lesson/2/2.2, n.d. Accessed: 2024-05-01.

[35] Philipp Probst and A. Boulesteix. To tune or not to tune the number of trees in random forest? *ArXiv*, abs/1705.05654, 2017.

[36] Sunil Ray. A complete tutorial which teaches data exploration in detail. https://www.analyticsvidhya.com/blog/2016/01/guide-data-exploration/, 2016. Accessed: 2024-05-03.

[37] B. Saket, Hannah Kim, Eli T. Brown, and A. Endert. Visualization by demonstration: An interaction paradigm for visual data exploration. *IEEE Transactions on Visualization and Computer Graphics*, 23:331–340, 2017.

[38] V. Sessions and M. Valtorta. The effects of data quality on machine learning algorithms. In *Proceedings of the 11th International Conference on Information Quality (ICIQ 2006)*, volume 6, pages 485–498, Cambridge, MA, 2006. MIT.

[39] Ozan Sonmez, Nezih Yigitbasi, Alexandru Iosup, and Dick Epema. Trace-based evaluation of job runtime and queue wait time predictions in grids. In *Proceedings of the 18th ACM International Symposium on High Performance Distributed Computing*, HPDC '09, page 111–120, New York, NY, USA, 2009. Association for Computing Machinery.

[40] Statsmodels Development Team. Statsmodels: Statistics in python. https://www.statsmodels.org/dev/index.html, 2024. Accessed: 2024-05-06.

[41] TechTarget. What is regression in machine learning? https://www.techtarget.com/searchenterpriseai/feature/What-is-regression-in-machine-learning, n.d. Accessed: 2024-04-30.

[42] Georgina M. Tinungki. The analysis of partial autocorrelation function in predicting maximum wind speed. *IOP Conference Series: Earth and Environmental Science*, 235, 2019.

[43] Carl Witt, Marc Bux, Wladislaw Gusew, and Ulf Leser. Predictive performance modeling for distributed batch processing using black box monitoring and machine learning. *Information Systems*, 82:33–52, 2019.

[44] Zabbix. Linux monitoring and integration with zabbix. `https://www.zabbix.com/integrations/linux`, 2023. Accessed: 2024-04-15.

[45] Zabbix. Features overview. `https://www.zabbix.com/features`, 2024. Accessed: 2024-04-29.

A Appendices

A.1 Data Quality - Consistency between Metrics

This showcases the inconsistency in the metric collections between the hosts.

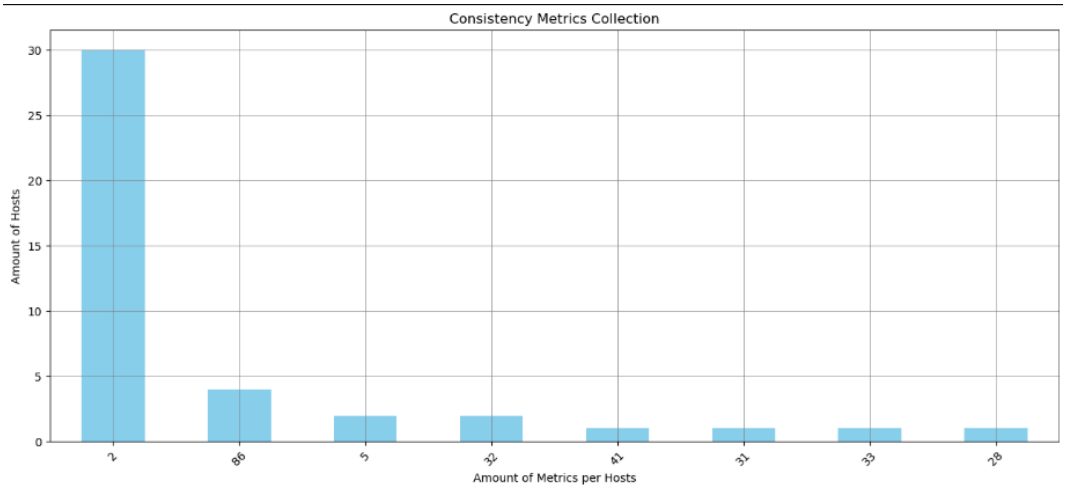


Figure 29: The Y-axis shows the number of hosts with the X-axis, amount of metrics. It explains how most hosts do not share the same amount of metrics, whereas most hosts only contribute with two metrics.

A.2 Time Variability

This showcases the inconsistency between hosts when collecting metrics for the whole period before the split. Some hosts have more gaps in data from collecting than others.

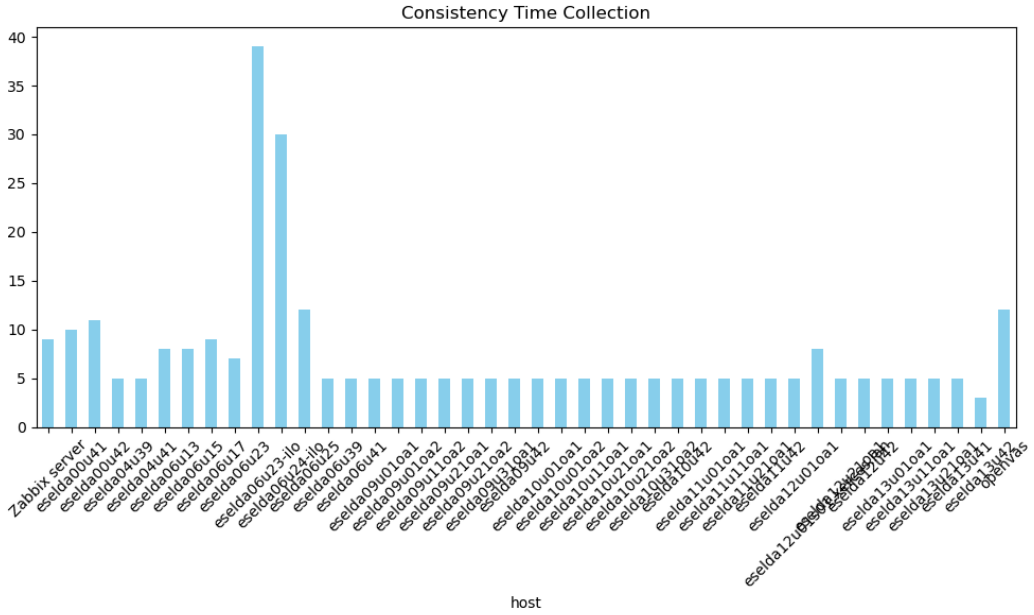
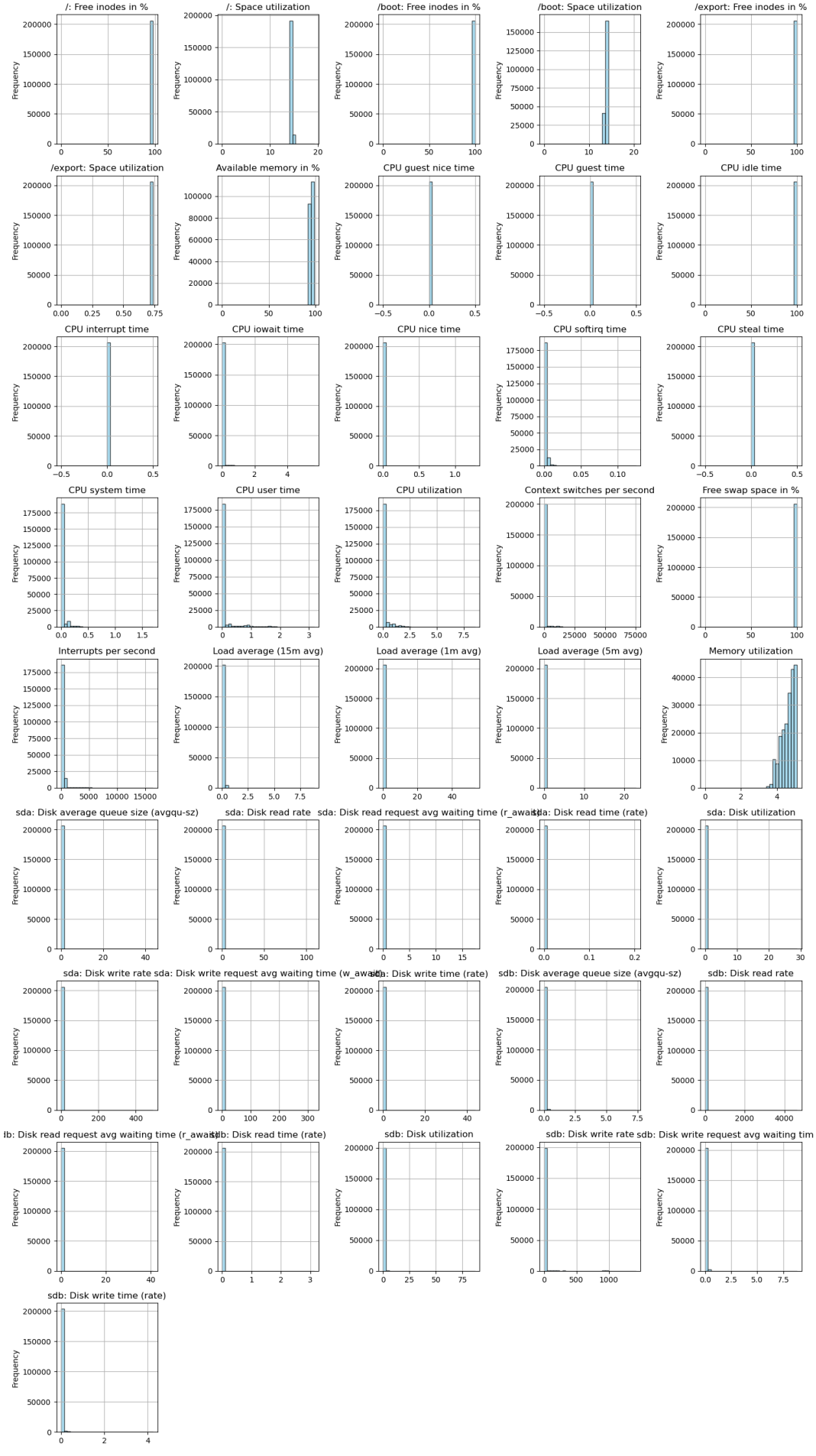


Figure 30: The Y-axis shows the variance in time for hosts. The X-axis shows hosts. The difference in variance collected between the metrics displays a small inconsistency.

A.3 Histogram of data



A.4 Weekly Trends

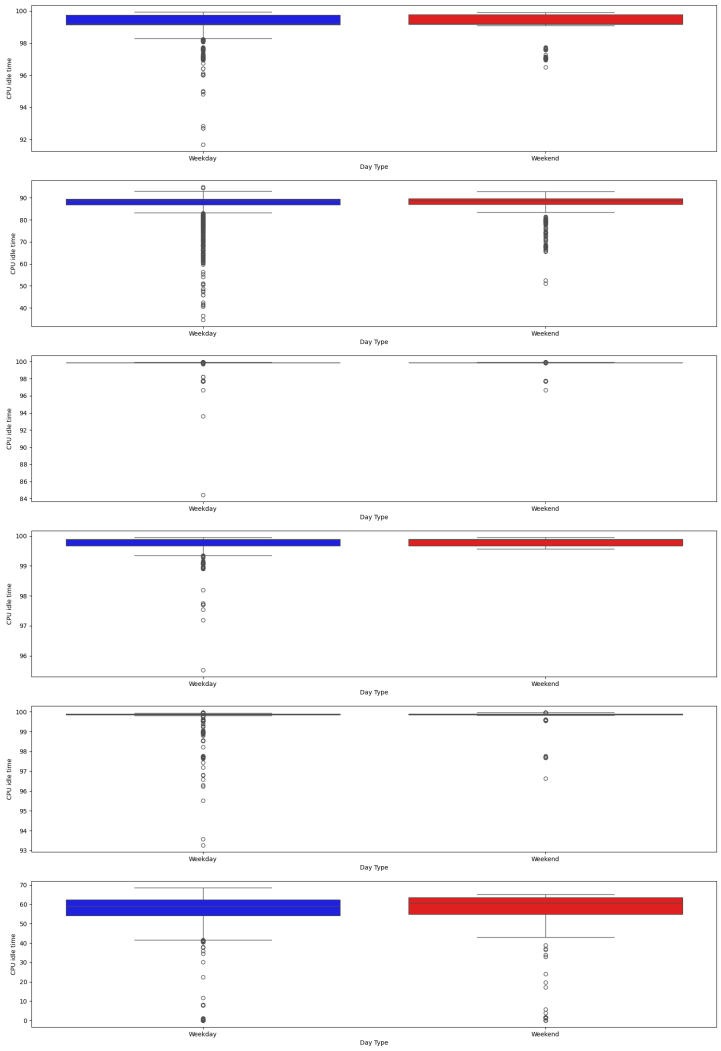


Figure 32: CPU activity shown between weekends and weekdays, where each plot represents a different Host. As it shows the metric CPU idle time, lower values correspond to higher activity. More outliers are shown during weekdays, which could be assumed to be spikes in CPU usage.

A.5 Data between all hosts from the established dataset

This showcases all hosts' metric activity for the established dataset.

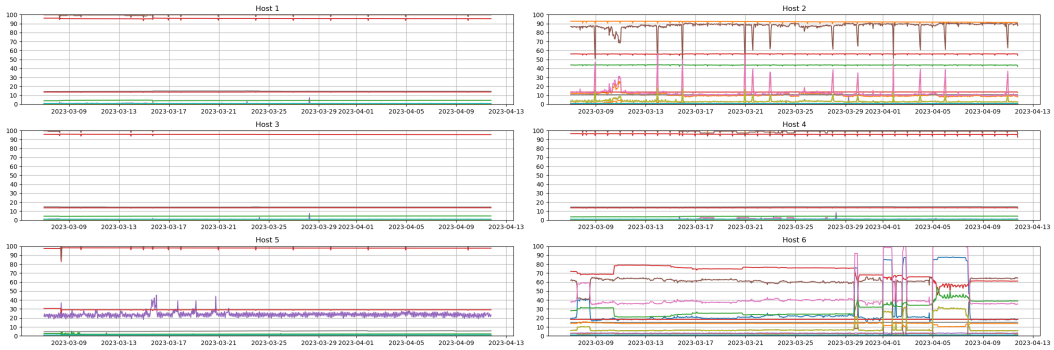


Figure 33: The Y-axis shows each line's value and metric activity, and the X-axis is the time period. This shows the real scale with no limit to the Y-axis of all host's metrics during the 34-day period.

Figure 34: The Y-axis shows each line's value and metric activity, and the X-axis is the time period. This shows the Y-axis limited scale to 100 of all host's metrics during the 34-day period.

A.6 PACF and ACF graphs 3D daily Resolution

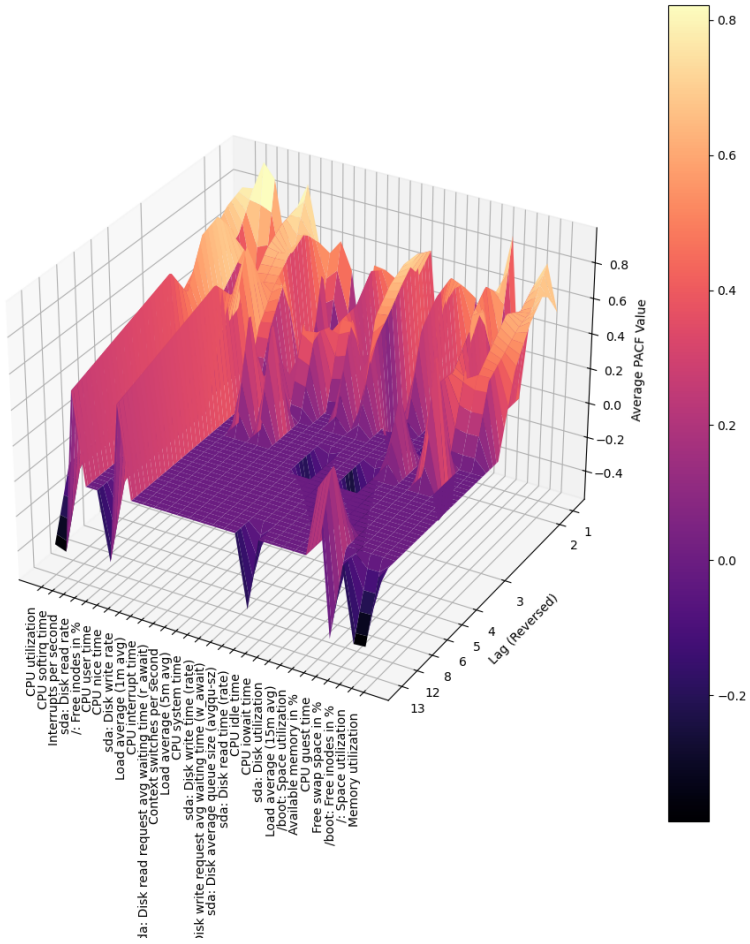


Figure 35: 3D graphs over PACF significant metrics, where the Y-axis is the value of each lag, the X-axis is the corresponding lag, and the Y-axis is the quantity of the responding metric.

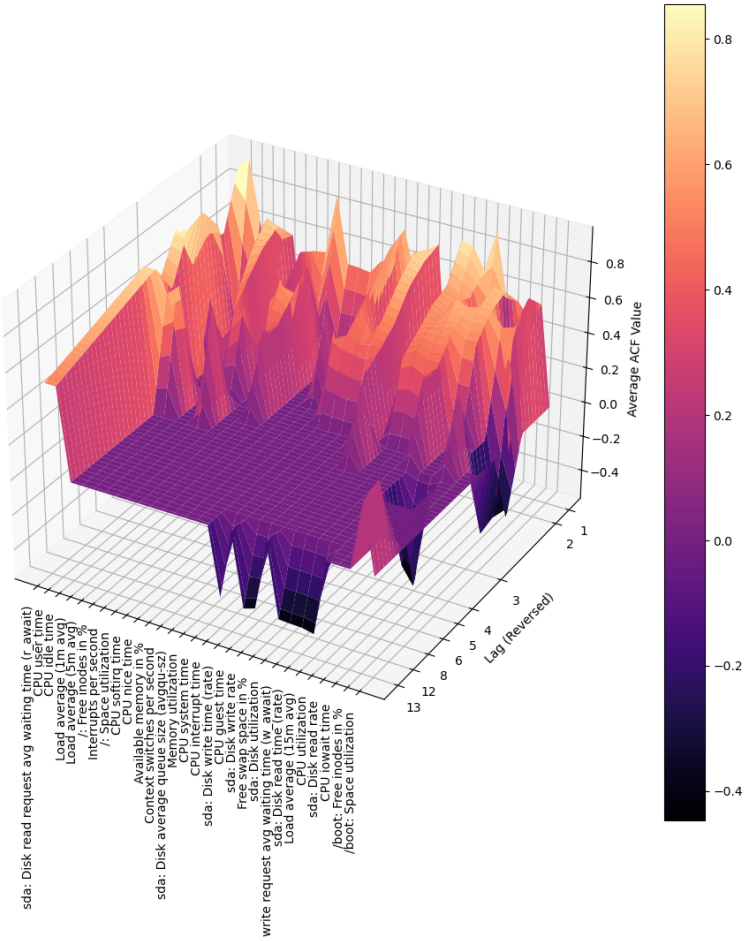


Figure 36: 3D graphs over ACF significant metrics, where the Y-axis is the value of each lag, the X-axis is the corresponding lag, and the Y-axis is the quantity of the responding metric.

A.7 PACF and ACF graphs 3D Hourly Resolution

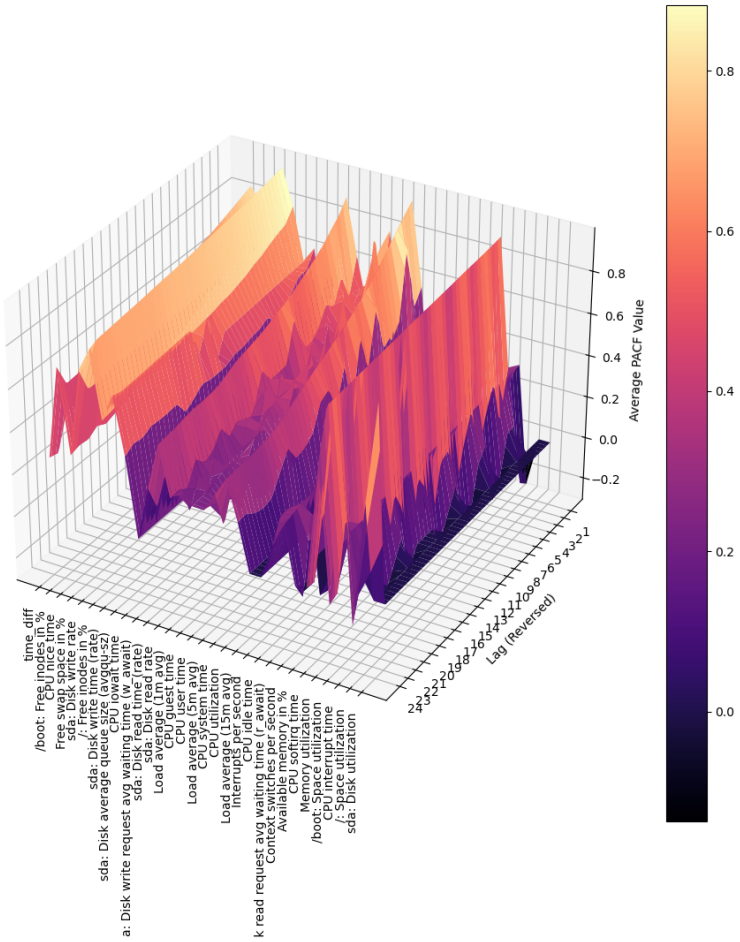


Figure 37: 3D graphs over PACF significant metrics, where the Y-axis is the value of each lag, the X-axis is the corresponding lag, and the Y-axis is the quantity of the responding metric.

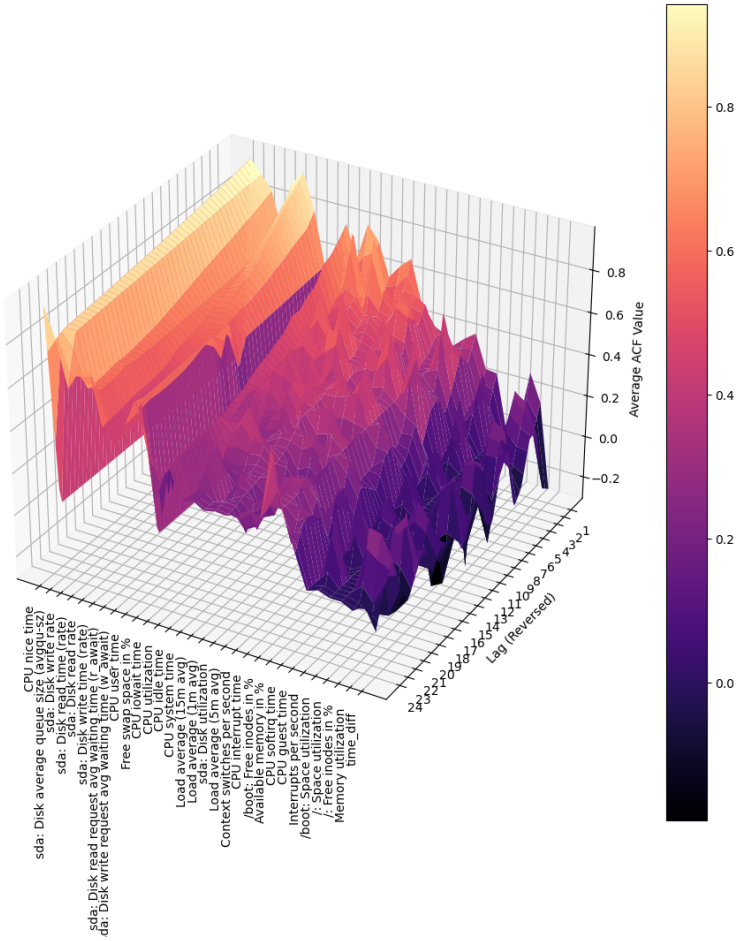


Figure 38: 3D graphs over ACF significant metrics, where the Y-axis is the value of each lag, the X-axis is the corresponding lag, and the Y-axis is the quantity of the responding metric.

A.8 Four weeks plot used for prediction

These are the weeks of data that the ML model was tested on.

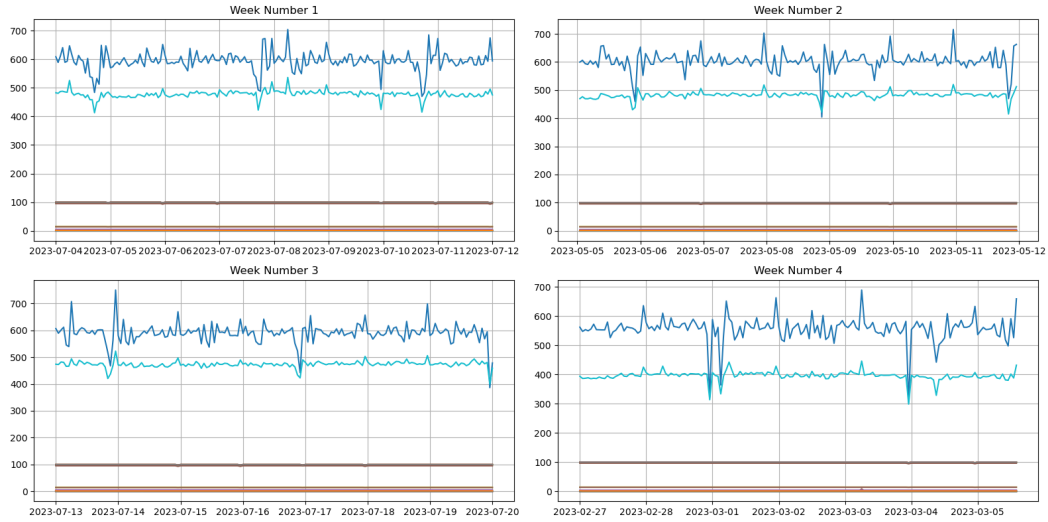


Figure 39: The Y-axis shows each line's value and metric activity, and the X-axis is the time period. A true scale of (Y-axis is not limited) the weeks that were used to perform predictions. Each plot represents one week.

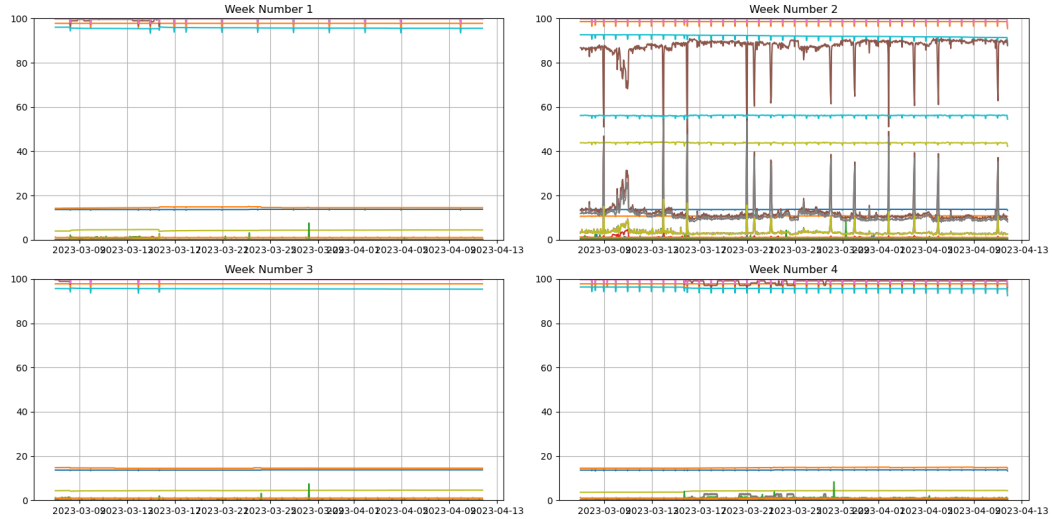


Figure 40: This shows the Y-axis limited scale to 100 of all weeks metrics, and the X-axis is the time period. Each plot represents one week.