Article

Machine Learning for Data Center Optimizations: Feature Selection Using SHapley Additive exPlanation (SHAP)

Yibrah Gebreyesus [a] , Damian Dalton [b], Sebastian Nixon [c], Davide De Chiara [d] and Marta Chinnici [e]

[a] School of Computer Science, University College of Dublin, Dublin, Ireland; yibrah.gebreyesus@ucdconnect.ie
[b] School of Computer Science, University College of Dublin, Dublin, Ireland; damian.dalton@ucdconnect.ie
[c] School of Computer Science, Wolaita Sodo University, Wolaita, Ethiopia; dr.nixon14@gmail.com
[d] ENEA-R.C. Portici, 80055, Portici (NA), Italy, davide.dechiara@enea.it
[e] ENEA-R.C. Casaccia, 00196, Rome, Italy, marta.chinnici@enea.it
* Correspondence: damian.dalton@ucdconnect.ie; Tel.: +353-87922-1156

Abstract: The need for Artificial Intelligence (AI) and Machine Learning (ML) technologies is increasingly being leveraged for optimizing Data centers' (DCs') operations as the volume of operations management data increase tremendously. These strategies can assist operators in better understanding their DC operations and making informed decisions up front to preserve service reliability and availability. Aiming at creating models that optimize energy efficiency, identify inefficient resource utilization and scheduling policies, and predict outages. Apart from model hyperparameter tuning, feature selection is a crucial step to identify relevant features with the objective of providing insight into the data, improving performance, and reducing computational expenses. Although several feature selection methods have been discussed in various domains, none have been discussed in the context of the data center. This paper introduces SHapley Additive exPlanation (SHAP), a class of additive feature attribution values-based feature selection that is rarely discussed in literature. We compared the effectiveness of SHAP method with several widely used methods. We used a real DC dataset obtained from the ENEA CRESCO6 cluster with 2,0832 cores to evaluate the methods. To demonstrate the comparison of the methods, we picked the top 10 most important features from each method, the predictions were retrained, and their performance was evaluated using MAE, RMSE, and MPAE. The results show that the SHAP-assisted feature selection performed best and align with human intuition.

Keywords: Data Center; Artificial Intelligence; Machine Learning; Feature Selection; SHAP; Game Theory

1 Introduction

A plethora of data-driven business processes, governmental and educational systems, as well as the rapid adoption of Industry 4.0 digital technologies like internet-enabled devices coupled with cloud computing platforms, the Internet of Things (IoT), Artificial Intelligence (AI) and Machine Learning (ML) technologies, big data streaming services, blockchain, robotics, and 3D technologies [1] [2], are accelerating the demand and complexity of data center (DC) industries. This trend has recently strengthened in the global fight against the COVID-19 pandemic and societal problems [3]. As data center demand and complexity increased, so is the operational management challenges, making it more difficult for operators to maintain service reliability and availability. Energy management is one of the most common and complex challenges. According to A. S. Andrae and T. Edler [4], if appropriate measurements are not taken, data centers are expected to consume up to 21% of global demand; if appropriate measurements are

 doi:10.20944/preprints202212.0482.v1

taken, this figure could be reduced to 8% by 2030. Hence, a new solution is expected to optimize DC operations and energy efficiency.

The emergence of IoT and intelligent technologies have recently enabled DC operations management to be automated by tracking operations parameters and generating massive amounts of data streams over time, allowing DC operators to make data-driven decisions. However, the data streams must be transformed into actionable information to optimize DC operations. So far, the most common methods for modeling DC operations and analyzing data streams were based on a heuristic, statistical, and mathematical models, which are reactive and incapable of processing massive amounts of data streams with complex and non-linear interactions [5]. Recently, artificial intelligence and machine learning technologies have been increasingly leveraged in the data center industries to model and process massive amounts of data streams into actionable information. Google has implemented a simple neural network ML approach for predicting Power Usage Effectiveness (PUE), which assisted in configuring controllable parameters and resulted in a 40% cooling efficiency [6]. Other research by A. Grishina et al., [7] was also conducted on thermal characterization and analysis using ML to enhance DC energy efficiency. Even though many more AI and ML-based research studies have been conducted to optimize DC energy efficiency and operations at different layers, relevant feature selection has been rarely discussed. However, relevant feature selection is the backbone of effectively modeling the target problem with the objective of improving model performance, reducing computational expense, and providing insights into the data streams. Although several feature selection methods have been discussed in various domains, as far as our knowledge, rarely been discussed in the context of the DC industry. Hence, identifying relevant features in the context of a data center is essential for mining the underlying patterns and effectively modeling the target problems.

This paper introduces Shapley Additive exPlanation (SHAP)-based feature selection method, which is a class of additive feature attribution values rarely discussed in literature. SHAP is a unique, consistent, and accurate additive feature attribution method based on the concept of Game Theory. It was initially proposed by Lundberg and Lee [8] for explainable AI (XAI) purposes in the field of AI. However, SHAP recently demonstrated a promising result in selecting relevant features by computing the importance of each feature for effectively modeling the target problem. It can also be applied to machine learning and deep learning models. Therefore, the ultimate goal of this paper is the identification of relevant features based on their importance computed using SHAP in relation to the target variables. We compared the effectiveness of SHAP assisted feature subset selection (FSS) method with several widely used feature selection methods. We used operational management data streams obtained from an HPC DC, the ENEA-CRESCO6 cluster, to demonstrate feature selection in the context of the data center. The data streams consist of energy consumption-related parameters, cooling-related parameters, and environmental-related parameters. DC energy demand represented as dcenergy, and ambient temperature, represented as amb temp, are the two chosen target variables to demonstrate the feature selection and analysis processes established in this paper.

The main contributions of this paper are: (i) Establishing an appropriate SHAPvalue-assisted feature automation approach in the context of a dynamically changing and complex DC environment with non-linear operating parameter interactions to identify relevant features for effectively modeling the target problem. It also helps DC operators better understand the relationships among operational parameters to accurately characterize DC operations. (ii) Understanding the underlying patterns and relationship amongst critical relevant features for accurate characterization of DC energy demand and ambient temperature. (iii) Demonstrating the performances of the different importance-based feature selection techniques established in the identification of relevant features in relation to the target variables. This paper also demonstrates how importance-based feature selections are effective in multivariate time series problems. (iv) Finally, identifying the best feature selection techniques that attempt to capture significant features, improve model performance, and reduce computational expense for DC operations characterization. We also demonstrated analyzing and characterizing a single feature dependency over the total sample to understand how a specific feature can impact the DC operation against a target variable.

 doi:10.20944/preprints202212.0482.v1

After that, the top 10 features of each method will be chosen, and these features will be used to retrain a machine learning model to predict energy demand and ambient temperature in order to evaluate and identify the best feature selection method. The main contribution of this paper is to identify and introduce essential feature selection methods in the context of data centers to effectively model and reduce computational expenses of machine learning models for DC operations characterizations. The remaining parts of this paper are organized as follows: Section 2 presents a theoretical review of feature selection techniques with a focus on techniques used in the FSS space, particularly in time series. Section 3 presents the methodology used in this paper. Section 4 presents experimental results and analysis, and Section 5 presents conclusions and future works.

2 Brief background and Theoretical Review of Feature Selection Techniques

In a given data stream **D** with **n** features, 2^n possible feature subsets can be generated. However, all these subsets may not be relevant for modeling and mining important patterns. Some features may appear to be equidistant, redundant, irrelevant, or noisy. To overcome these challenges, there are two special methods used to identify relevant features [9]. The first one is feature extraction/dimensional reduction, which transforms the original input feature into a reduced representation set, while the second one is feature selection, which identifies relevant subsets while preserving the original information [10][11]. Hence, in this paper, we focus on feature selection methods to identify relevant features in the context of DC. The main classes of feature selection (FS) methods are wrapper methods, embedded methods, or filter methods [12]. Wrapper methods use the model's performance as a score to select relevant feature subsets [13]. Even though wrappers are effective methods, computationally they are expensive and prone to over-fit [14]. On the other hand, embedded feature selection methods are applied during the model training process and are associated with a specific learning algorithm [11][12]. Filter methods are also another model-independent feature selection method typically applied in the preprocessing steps [15].

However, the best strategy usually depends on the task at hand. The data center operational management data streams are multivariate time series problems. The data streams are sequences of observations denoted as $x_i(t);[i = 1,...,n;t = 1,...,T]$ [16]. Where x represents observations, i represents measurements taken at each time point t, n is the maximum index, and T is the maximum time length. In general, the time series may be Univariate Time Series (UTS) when n is 1 and Multivariate Time series (MTS) when n is greater or equal to 2. The DC operations management data streams are MTS data streams stored as a $m * n$ matrix, where m represents the number of observations and n represents the number of features or variables. The interaction between MTS data streams over time is the key complexity of the MTS data streams. Thus, feature subset selection (FSS) entails identifying relevant features from given data streams with three goals in mind. The goals are to provide insights into data, reduce computational costs, and improve model performance [12]. To achieve these objectives, many studies have been conducted to identify relevant features in classification and regression problems. Wrapper methods such as recursive feature elimination (RFE) and backward feature elimination (BFE) selection methods, require each item to be input in the form of a column vector. However, MTS data streams are stored in the form of a matrix, wrapper methods are unsuitable for MTS problems because correlation-related information between features may be lost when vectorizing. MTS data streams typically contain complex correlations between features over time.

Filter methods are correlation-based feature selection methods that are effective on time series or continuous variables and calculate correlations among different features and the target variable. Using correlation matrices such as Pearson, Spearman, and Kendall, filter methods identify relevant features with a high correlation to the target variable. The filter model selects relevant subsets of the input variables based on distance, dependency, and consistency [17]. Commonly used correlation-based methods are Pearson, Spearman, and Kendall, as well as the Mutual Information (MI) method. The Pearson correlation is the most widely used filter method for measuring the linear relationship between two variables. The Spearman and Kendall correlation methods, which employ non-parametric tests, are better suited for non-normally distributed data [18]. The degree of correlation between two variables is measured by the Spearman correlation, whereas the Kendall correlation, an extension of Spearman, measures the strength of dependence between two features [19]. There have been various claims that the Kendall correlation provides more accurate data generalization than Spearman features [19].

 doi:10.20944/preprints202212.0482.v1

Spearman correlation can be effective for non-linear and non-time series data but shows poor results in the domain of MTS problems. A study also implemented Pearson's correlation and symmetrical uncertainty scores together to compute linear and non-linear relationships between features and target variables [20]. In this context, a correlated but important feature as well as a feature that has transitivity impacts on other features may be overthrown and lead to the wrong conclusion. Another well-known feature selection method is Mutual Information (MI) based method, which measures the uncertainty of random variables, termed Shannon's entropy [21]. A recent feature-subset selection method based on merit score, also implemented by Kathirgamanathan and P. Cunningham [22], was used to identify relevant features in the MTS domain. In general, correlation-based feature selection (CFS) techniques have been successfully used outside of the time series domain to select a feature subset from multivariate data [23]. However, it is poor in the time series domain because it requires data in a feature vector representation, which loses important features during vectorization in time series.

Recently, importance-based feature selection methods have been used in different domains, including the MTS domain. Commonly used methods are Random Forest (RF) and Xtream Gradient Boosting (XGB)-based feature importance ranking methods. Zhen Yang et al. [24] for example, used random forest with Gini Feature Importance ranking to identify relevant features related to PUE prediction. However, these methods suffer from a high frequency and cardinality of features. This paper establishes SHAP values assisted relevant feature selection for effectively modeling and characterizing DC operations.

3 Methodology

This paper focuses only on supervised learning approaches. Because modeling DC energy demand and ambient temperature predictions are treated as regression problems. We apply SHapley Additive exPlanation (SHAP) additive feature attribution values assisted feature selection method to identify relevant features based on various feature importance rankings to effectively model the specified target problems. Subsection 3.2 presents the description and implementation procedures of SHAP. In feature importance-based methods, input features are assigned a score based on how useful they are at predicting target variables. The top-ranked features are the most significant features for modeling the specified target problem. We compared the effectiveness of the SHAP-values assisted method with several commonly used importance-based feature selection methods. These are; (i) Random Forest with Gini Feature Importance Ranking (RFGFIR) (mean decrease impurity), (ii) Random Forest Permutation based Feature Importance Ranking (RFPFIR), (iii) Random Forest with SHAP Values based Feature Importance Ranking (RFSVFIR), (iv) Extreme Gradient Boosting with Gain Feature Importance Ranking (XGBGFIR), (v) Extreme Gradient Boosting with Permutation based Feature Importance Ranking (XGBPFIR) and (vi) Extreme Gradient Boosting with SHAP Values based Feature Importance Ranking (XGBSVFIR). The methods were experimented using historical data obtained from an HPC data center, the CRESCO6 cluster, by splitting samples into a 7:3 for training and testing, respectively, preserving the time order. Table 1 presents the data set and descriptions. We implemented testing sets to compare the different importance-based feature selection methods applied in this paper. Then, we retrained and compared the models' performance and learning rate using the top 10 ranked features of each feature selection method. We used mean absolute average error (MAPE), mean absolute error (MAE), and root mean squared error (RMSE) evaluation criteria to evaluate the models' performance, which is commonly used in time series regression problems. The method with the lowest error and computational expenses is selected as the best method for the identification of relevant features for effectively modeling DC operations. In this paper, we generally adhere to the conceptual framework depicted in Figure 1. The following three steps have been systematically applied: (i) data sets and preprocessing described in subsection 3.1, (ii) introduction to SHAP feature attribution values based relevant feature selection and analysis describes in subsection 3.2, and (iii) machine learning models discusses in subsection 3.3.

 doi:10.20944/preprints202212.0482.v1

3.1 Datasets and Descriptions

The dataset used in this paper is obtained from the ENEA CRESCO6 cluster, a High-Performance Computing (HPC) data center consisting of 434 computing nodes with a total of 20,832 cores. Each node is equipped with 2 Intel Xeon Platinum 8160 CPUs (represented as CPU1 and CPU2), each with 24 cores and a total of 192 GB of RAM, corresponding to 4 GB/core and operating at a clock frequency of 2.1 GHz. ENEA CRESCO6 cluster has been operating since 2018. The cluster has a nominal computing power of 1.4 PFLOPS and is based on the Lenovo Think System SD530 platform. The nodes are interconnected by an Intel Omni-Path network with 15 switches of 48 ports each at 100 Gb/s with latency equal to 1 μs bandwidth. For monitoring and management purposes, each computing node of CRESCO6 are equipped with onboard sensors. These sensors could read vital and non-vital parameters of the hardware for the entire calculation node. These sensors detect various temperatures at different points of the calculation node, particularly CPU and RAM, fans' rotation speeds, the volume of air that passes through the node, and energy parameters that provide the state of energy consumption each time it is invoked.

The data is read via intelligent platform management interfaces (IPMI) which is a CONFLUENT software package directly installed on the cluster computing nodes and stores the values acquired in a MySQL database. Cooling system parameters like inlet temperature, outlet temperature, relative humidity, airflow, and fan speed are monitored using onboard sensors of the refrigerating machine. There are also several sensors installed around the cluster to monitor the environmental operating conditions of the cluster in the data center, which measure temperature and humidity-related parameters. In this paper, we used 2020 annual data from three tables in the MySQL database, which consists of energy consumption, cooling, and environmental data streams. These data streams are read in a matter of seconds or minutes intervals. The datasets were organized, standardized, sensitized, and missing values were interpolated. So, we resampled the dataset into 15 minutes intervals equal in length, which is a resealable number of minutes that the DC operators require to respond to events in the data center operating environment and to align all the available tables of data that compose the dataset. Finally, the datasets were aggregated and shaped as (35136, 50), that is, 35136 draws and 50 features. The following time covariate also considers features with periodic behaviors, these are hours, days, weekdays, weekends, months, and quarters of the year, which could improve the modeling performance of the target variable. Table 1 describes the features and their descriptions used in this paper.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 December 2022 doi:10.20944/preprints202212.0482.v1

Table 1: Features and their descriptions

No	Feature Name	Description
1	Timestamp_measure	Datetime index
2	sys_power	Total instantaneous Power Measurement of commuting node (watt)
3	cpu_power	CPU power measurement of the computing node (watt)
4	Mem_power	Ram memory power measurement of the computing node(watt)
5	fan1a	Speed of fan represented as fan1a installed in the node expressed as RPM (rev per minute)
6	fan1b	Speed of fan represented as fan1b installed in the node expressed as RPM (rev per minute)
7	Fan2a	Speed of fan represented as fan2a installed in the node expressed as RPM (rev per minute)
8	fan2b	Speed of fan represented as fan2b installed in the node expressed as RPM (rev per minute)
9	fan3a	Speed of fan represented as fan3a installed in the node expressed as RPM (rev per minute)
10	fan3b	Speed of fan represented as fan3b installed in the node expressed as RPM (rev per minute)
11	fan4a	Speed of fan represented as fan4a installed in the node expressed as RPM (rev per minute)
12	fan4b	Speed of fan represented as fan4b installed in the node expressed as RPM (rev per minute)
13	fan5a	Speed of fan represented as fan5a installed in the node expressed as RPM (rev per minute)
14	fan5b	Speed of fan represented as fan5b installed in the node expressed as RPM (rev per minute)
15	sys_util	Percentage of use of the system (%)
16	cpu_util	Percentage of use of CPU's of the computing node (%)
17	mem_util	Percentage of use of the RAM memory of the computing node
18	io_util	Node i/o traffic (mb)
19	cpu1_Temp	CPU1 temperature (°C)
20	cpu2_Temp	CPU2 temperature (°C)
21	sys_airflow	System air flow of nodes measured in cubic feet to minute (CFM)
22	exh_temp	Exhaust temperature that is air exit of the nodes in (°C)
23	amb_temp	A temperature near the computing nodes or DC room operating temperature (°C)
24	dcenergy	Data energy demand meter consumed up to the next reading
25	supply_air	Cold air/inlet temperature (°C) that blows from CRAC through vented floor to remove hot air
26	return_air	Ejected heat or warm air back from racks to the outside (°C)
27	relative_umidity	Working humidity of the CRAC (°C)
28	fan_speed	The speed of the cooling system fan within the CRAC to regulate airflow rate within the DC (RPM)
29	cooling	Cooling working intensity of the DC (%)
30	free_cooling	Not applicable / values presented as 0
31	hot_103_temp	Environmental hot temperature sensor installed around the computing node (°C)
32	hot103_hum	Environmental humidity temperature sensor installed around the computing node (°C)
33	hot101_temp	Environmental hot temperature sensor installed around the computing node (°C)
34	hot101_hum	Environmental humidity temperature sensor installed around the computing node (°C)
35	hot111_temp	Environmental hot temperature sensor installed around the computing node (°C)
36	hot111_hum	Environmental humidity temperature sensor installed around the computing node (°C)
37	hot117_temp	Environmental hot temperature sensor installed around the computing node (°C)
38	hot117_hum	Environmental humidity temperature sensor installed around the computing node (°C)
39	hot109_temp	Environmental hot temperature sensor installed around the computing node (°C)
40	hot109_hum	Environmental humidity temperature sensor installed around the computing node (°C)
41	hot119_temp	Environmental hot temperature sensor installed around the computing node (°C)
42	hot119_hum	Environmental humidity temperature sensor installed around the computing node (°C)
43	cold107_temp	Cold temperature (°C)
44	cold107_hum	Cold humidity (°C)
45	cold105_temp	Cold temperature (°C)
46	cold105_hum	Cold humidity (°C)
47	cold115_temp	Cold temperature (°C)
48	cold115_hum	Cold humidity (°C)
49	cold113_temp	Cold temperature (°C)
50	cold113 hum	Cold humidity (°C)

The above-mentioned features represent totals and averages, which are derived from each sensor's data. In addition to the features listed above in Table 1, we considered hours, weekdays, weekends, months, and quarters of the year, which have a significant impact on modeling the time series problems. Hence, the applications of the feature selection methods implemented in this paper are shaped as (35136, 56).

3.2 Introduction to SHapley Additive exPlanation (SHAP)

SHapley Additive exPlanation (SHAP) is one of the Additive Feature Attribution methods initially proposed by Lundberg and Lee (2017), which was designed for explainable AI (XAI) [8]. The explanation level is focused on comprehending how a model

 doi:10.20944/preprints202212.0482.v1

makes decisions based on its features and from each learned component. SHAP is a class of additive feature attribution methods that are model-agnostic and can be applied to any machine learning and deep learning models by attributing each input feature's importance. In comparison to other additive feature attribution methods like LIM [25], DeepLIFT[26], and others, SHAP has a unique and satisfies accuracy, missingness, and consistency properties. In SHAP, feature importance can be computed using ideas from the Game-Theory concepts. The model explanation can be computed at global and local levels. The model's global explanation assists in better understanding which features are important and the interactions between features. It is also more aligned with human intuition and more effective at mining influential features. Hence, this paper introduces SHAP values-assisted feature selection to identify relevant features with respect to the specified target variables in the context of data center.

SHAP-assisted feature selection procedures are depicted in Figure 1 as; (i) Data collection and preprocessing; (ii) In the initial interaction, all features are used to train the models; (iii) SHAP is then applied, and then computed Shapley values of each feature and ranked in ascending priority order; (iv) Then, train the models *n* times, beginning with the topmost features and continuing until the optimal subset is found with respect to each of the aforementioned methods; and (v) Finally, pick the optimal feature subset to effectively model the target problem with the most predictable feature subset to optimize DC operations. In general, Shapley values can be computed as a unified measure of feature importance, which is the average of the marginal contributions of features from all conceivable coalitions. For example, we can compute the Shapley value of a given feature **n** in a given dataset **D**, the value of the feature is replaced by a value from another instance of the model, and all possible outcomes are considered to compare the original prediction with the new prediction. Hence, the average between the new value and the original value represents the importance of feature *n* to the final prediction. For example, Shapley value estimation of the j^{th} feature with a number of iterations *i*, an instance of interest *x*, feature index *j*, dataset *D* with matrix *X*, and predictive model *f*, defined as:

$$\phi ij = f\hat{}(x+j) - f\hat{}(x-j), \quad (1)$$

where ϕ^i_j is the average Shapley value of *jth* feature with *i* iteration, $f\hat{}(x_{+j})$ is the prediction for interest x with a random number of feature values including *jth* feature, $f\hat{}(x_{-j})$ is coalition without *jth* feature. In general, to compute the Shapley value of *jth* feature in the interest of *x* it is given as below:

$$\phi_j(x) = \frac{1}{n}\sum_{i=1}^{n}(\phi_j^i) \quad (2)$$

The importance of all features is computed in the same way and ranked based on their Shapley value in a prior order. As shown in the following iterations, the model is then trained using the most important features, beginning from the top and continuing until the optimal feature subset is found.

1. 1^{st} interaction: $i_1 = f_1$
2. 2^{nd} interaction: $i_2 = f_1, f_2$
3. n^{th} interaction: $i_n = f_1, f_2, f_3,...,f_n$

Finally, the optimal feature subset can be found and used to effectively model the target problem.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 December 2022 doi:10.20944/preprints202212.0482.v1

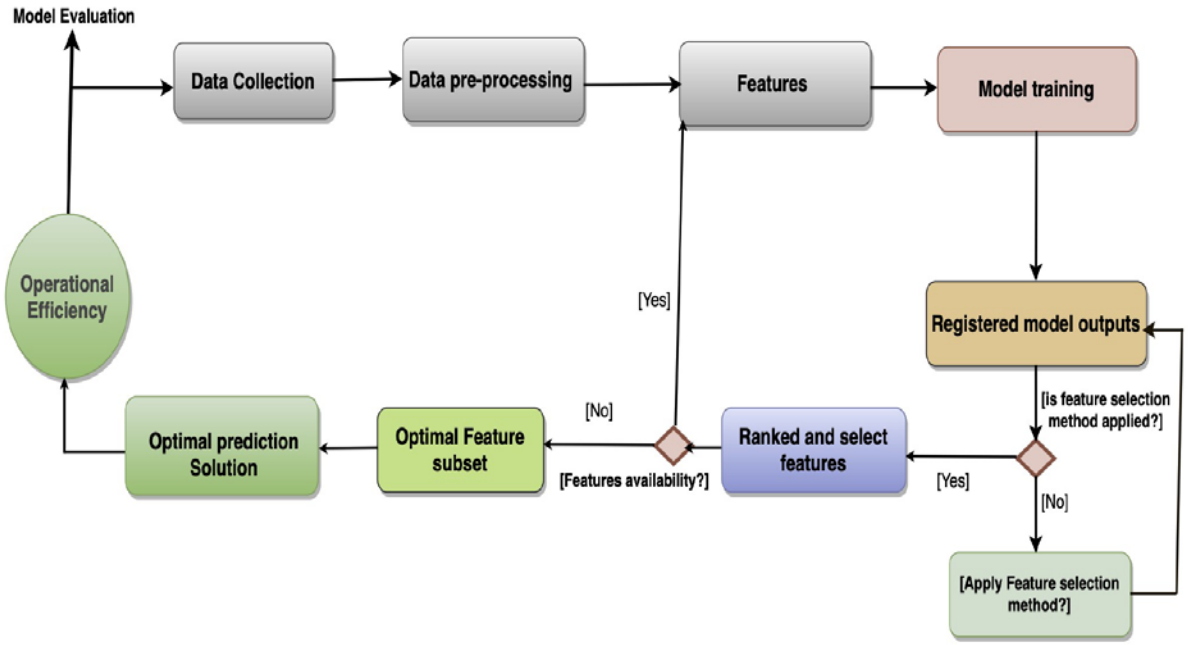


Figure 1: The flow of SHAP assisted FSS method. Relevant feature selection and analysis processes in the context of the DC operations. The colors represent different tasks.

SHAP has the following approximation method (Kernel SHAP) and three specific approximation methods for the model type (Gradient SHAP, Tree SHAP, and Deep SHAP). Kernel SHAP improves the sample efficiency of estimates of SHAP values without considering the model type. By restricting ourselves to the specific model type, such as Deep, Gradient, and TreeSHAP, faster approximation methods are obtained. As a result, in this paper, we used TreeSHAP in conjunction with the RF and XGB machine learning models to demonstrate relevant feature selection.

3.3 Machine Learning Models

Machine learning and deep learning techniques have recently been used to model and optimize data center operations. To effectively model the specified problem, identifying relevant features is key. Hence, this paper demonstrates importance-based relevant feature selection implemented with RF and XGB-supervised time series prediction models. The following subsections 3.3.1 and 3.3.2 describe the model's implementation.

3.3.1 Random Forest (RF)

Random Forest (RF) is an ensemble machine learning algorithm that is widely used in classification and regression problems. It's general framework was initially proposed by Ho (1995) [27] and further extended by L. Breiman (2001) [28]. The algorithm training is based on bagging, which is short for bootstrap aggregation. Each decision tree is trained on a data set drawn at random from the training data with replacement. In this case, each tree learner is shown a different subset of the training data, and the same observation can be chosen more than once in a sample [29]. In general, the algorithm follows the following steps: (i) In RF n number of random samples are generated from the given data set having k records. (ii) Individual decision trees are constructed for each sample. (iii) Then each decision tree generates sequential output. (iv) The final output is considered as majority voting or average for classification and regression respectively. Since the problem we have at hand is an MTS regression problem, we fit several decision tree classifiers on various sub-samples of the data set and then average the predictions. This could improve

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 December 2022 doi:10.20944/preprints202212.0482.v1

the predictive accuracy and avoid over-fitting of the RF regression model. During model implementation, the values of hyperparameters typically have a significant impact on the performance and behavior of the model. Hence, the hyperparameters for RF are explored and tuned as follows: The maximum depth of trees is 5, and the estimator, or forest, in the tree is 200, with the rest remaining as default.

3.3.2 Extreme Gradient Boosting (XGBoost)

Extreme Gradient Boosting (XGBoost) is a decision tree-based ensemble machine learning algorithm that uses a gradient boosting framework [30] with regularization to objectively reduce variance and bias. It is a scalable, Distributed Gradient Boosted Decision Tree (GBDT) machine learning library. It is also more effective and efficient implementation of the gradient boosting widely used ML algorithms in both classification and regression problem-solving. It has both linear model solver and tree learning algorithms. So, what makes it fast is its capacity to do parallel computation on a single machine. This makes XGBoost at least 10 times faster than existing gradient-boosting implementations [9]. It supports various objective functions, including regression, classification, and ranking. It also has additional features for doing cross-validation and finding important variables [30]. In regression problems, although it requires that the time series data be transformed into supervised data. The hyperparameters for XGB are explored and tuned as follows: The maximum depth is 5, the estimator is 100 and the learning rate is 0.01.

3.4 Criteria for model evaluation

The model error in model regression analysis is the difference between the actual data points and the best fit line produced by the algorithm. With multiple data points, the model's error will be determined using the following criteria.

(i) Mean Absolute Error (MAE): This is the average of the absolute values of the errors that represent the deviation from true probability. This is expressed mathematically as:

$$MAE = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{n}, \quad (3)$$

(ii) Root Mean Squared Error (RMSE): Because it can be interpreted as the standard deviation of the prediction errors, RMSE is a popular performance evaluation metric for models. This is computing as follows:

$$RMSE = \sqrt{(\frac{1}{n}) \sum_{i=1}^{n} (y_i - y_i)^2} \quad (4)$$

$$MAPE = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{n} * 100 \quad (5)$$

Execution time: This is when it takes the model to learn and make predictions from the input features.

Where $\hat{y}$ is the predictive value at time point i and y_i is the actual value. The lower the MAE, RMSE and MAPE values, the better the model performed in energy demand and ambient temperature predictions.

4 Results and discussion

The SHAP-assisted feature selection method established in this paper has been discussed and compared to several importance-based relevant feature selection methods. As a result, it is time to decide which of them is best suited for identifying relevant features for effectively modeling data center operations, allowing operators to perform data-driven service reliability and availability

 doi:10.20944/preprints202212.0482.v1

improvements. The feature importance is computed and scored for all input features for a given machine learning model. A higher score indicates that the specific feature contributes more to the problem model's effectiveness and efficiency. To demonstrate the feature selection process, we selected two widely used machine learning regression models. These regression machine learning models are Random Forest (RF) and Extreme Gradient Boosting (XGB). In machine learning models, the values of hyperparameters typically have a significant impact on the models' performance and behavior. Hence, the hyperparameters for each method are explored and tuned as follows. For RF-based feature selection methods, the hyperparameters are tuned as follows: The maximum depth of trees is 5, and the estimator, or forest, in the tree is 200; the rest is left as default. For the XGB, we set the hyperparameters as follows: The maximum depth is 5, the estimator is 100, and the learning rate is 0.01; the rest is left as default. All the feature selection methods are fitted and computed in Python.

For the first time, we trained the models using 70% of the data set and all the input features, then computed the importance of each feature and ranked them in ascending order. The most important feature receives the highest ranking. After computing and ranking features based on their importance using each feature selection method, we trained the models, starting with the highest-ranked features and working our way down to n -1 to find the optimal feature subset using each method. Feature ranking and the combination of optimal subsets of each feature selection method may vary due to the architecture and nature of the models. To evaluate the methods, we implemented a testing set, which is 30% of the total data set. We used Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) evaluation criteria to compare and evaluate the effectiveness of each feature selection method. For comparison purpose, we selected the top 10 most important features of each feature selection method and retrained the model. The results presented in Table 2 and Table 3 are based on the top 10 most important features for each of the feature selection methods. The results demonstrate which feature selection method is best suited to identify relevant features with respect to the specified target variable in the context of operational data center. Note that, in this paper, we have chosen data center ambient temperature and energy demand as target variables. Hence, the results presented in Table 2 and Table 3 are with respect to data center ambient temperature and energy demand, respectively.

Table 2: Experimental results for considering forecasting models. Data center ambient temperature (represented as amb_temp) is the target variable. FSS performance evaluation is based on the top 10 features of each method.

Feature Selection Methods	Evaluation metrics and execution time			
	MAE	RMSE	MAPE	Execution time (sec)
RFGFIR	0.644	0.512	0.033	102.011
RFPFIR	0.499	0.392	0.022	175.515
RFSVFIR	**0.42**	**0.237**	**0.018**	**120.790**
XGBGFIR	0.43	0.339	0.022	118.790
XGBPFIR	0.443	0.348	0.025	123.364
XGBSVFIR	**0.411**	**0.245**	**0.004**	**112.890**

Table 3: Experimental results for considering forecasting models. Data center energy demand (represented as dcenergy) is the target variable. FSS performance evaluation is based on the top 10 features of each method.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 December 2022 doi:10.20944/preprints202212.0482.v1

Feature Selection Methods	Evaluation metrics and execution time			
	MAE	RMSE	MAPE	Execution time (sec)
RFGFIR	14.241	22.112	0.032	115.011
RFPFIR	3.714	8.504	0.018	163.283
RFSVFIR	**1.368**	**6.657**	**0.005**	**142.830**
XGBGFIR	2.329	8.413	0.024	121.785
XGBPFIR	0.443	0.348	0.015	128.845
XGBSVFIR	**0.411**	**0.245**	**0.004**	**117.687**

Table 2 and Table 3 demonstrate the performance and computational expenses of the models with the top 10 most important features obtained from each feature selection method. The methods presented in the Tables are RFGFIR, RFPFIR, RFSVFIR, XGBGFIR, XGBGFIR, and XGBSVFIR. RFGFIR is computed during the model training process, which makes it computationally fast but error prone. On the other hand, RFPFIR can be computed based on the feature importance of permuted out-of-bag (OOB) samples based on decreasing mean model accuracy. It requires a trained model and test data to compute the feature importance. This method randomly shuffles each feature and computes the changes in the model's performance. The feature that has the greatest impact on model performance is the most important for effectively modeling the target problem. Hence, as demonstrated in Table 2 and Table 3, RFPFIR performed best compared with RFGFIR but is computationally expensive. XGBGFIR and XGBPFIR are two other commonly used importance-based feature selection methods. XGBGFIR can be computed during the model training using the feature importance _ attribute. Its value is computed as the average gain across all splits where the feature was used in the decision tree. Like RFPFIR, XGBPFIR requires a trained model and test data to compute the feature importance. This method randomly shuffles each feature and computes the changes in the model's performance. As the result demonstrated in both Table 2 and Table 3, XGBPFIR performs well compared with XGBGFIR but is computationally expensive. However, the above methods suffered from a high feature frequency and cardinality, which leads to wrong conclusion.

The SHAP-values-assisted FSS method introduced in this paper, on the other hand, outperformed others to identify relevant feature subset selection. Particularly, TreeSHAP was applied to RF and XGB models. Table 2 and Table 3 demonstrated that RFSVFIR and XGBSVFIR performed best with lower errors and fair speed. RFSVFIR performed with lower errors of MAE 0.42, RMSE of 0.227, MAPE of 0.018, and MAE of 1.368, RSME of 6.657, and MAPE of 0.0032 at predicting the DC ambient temperature and DC energy demand target variables, with fair computational speed, respectively. Similarly, XGBSVFIR performed with lower errors of MAE 0.411, RMSE of 0.245, MAPE of 0.004 and MAE of 0.364, RSME of 5.321, and MAPE of 0.0032 at predicting the DC ambient temperature and DC energy demand target variables, with fair computational speed, respectively. When we compare XGBSVFIR and RFSVFIR, XGBSVFIR performed best, with better performance and fair computational expenses. SHAP with XGB is faster than with RF due to the capacity for parallel computation in XGB. Furthermore, due to its optimized hyperparameter, TreeSHAP [31] accelerates computation when used with XGB. Hence, XGBSVFIR performs best with good model performance and fair computational speed compared to other methods as presented in Table 2 and Table 3. The SHAP-assisted method has the following main advantages over the others: (i) It is a unique, accurate, and consistent method of computing feature importance based on each feature contribution using a Game Theory approach. The degree of importance indicates the extent to which the explanation reflects the significance of the feature. The SHAP values calculated the importance of a feature by comparing what a model predicts with and without each feature. (ii) SHAP is model-agnostic and can be applied with machine learning and deep learning techniques. (iii) It is also aligned with human intuition, and more effective in mining influential features.

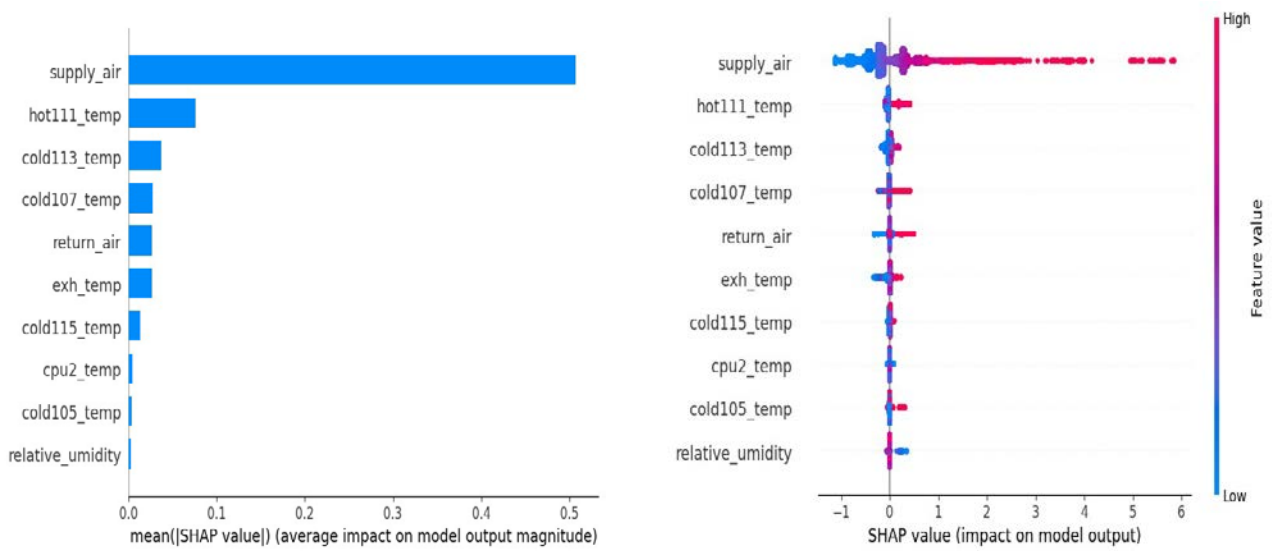
We rely on graphs to visualize the feature importance obtained from SHAP. The SHAP summary plots allow us to visualize the effect of the features on the total samples by plotting the SHAP values of each feature for each sample. In

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 December 2022 doi:10.20944/preprints202212.0482.v1

Figure 2.a and Figure 3.a features are represented by the sum of the magnitudes of the SHAP values in all the samples, i.e., by their $\sum_{i=1}^{n} |\phi_j^i|$

SHAP values are used to show the distribution of the impacts that each feature has on the model target outputs. Figures 2.a and 2.b are demonstrated with respect to DC ambient temperature target variable and Figures 3.a and 3.b are demonstrated with respect to DC energy demand. The SHAP values ϕ_j^i are drawn horizontally, stacked vertically when it runs out of space. Each point represents a row of the data set. The gradient color indicates the original value of that feature (high red, low blue). If the impact of each feature on the model output varies smoothly as its value changes, then this color will also have a smooth gradation. Also, we can just take the average absolute value of the SHAP values for each feature to get a standard bar graph, as presented in figures 2.b and 3.b. The vertical axis indicates the feature name with the importance from top to bottom priority order. On the horizontal axis is the SHAP values that indicates how much is the change in target variable. Figures 2.a and 3.a illustrate the average absolute value of the SHAP for each feature to obtain a standard bar graph representation with respect to the target variables. The vertical axis displays the feature names in ascending order of features importance from top to bottom. The horizontal axis contains the SHAP values, which show how much has changed over time.

In Figures 2.b and 3.b, features are classified by the sum of the magnitudes of the SHAP values in all the samples, i.e., by their global impact. SHAP values are used to show the distribution of the impacts that each feature has on the model output. SHAP values are drawn horizontally and stacked vertically when they run out of space. Each point represents a row of the data set. The gradient color indicates the original value of that feature (high red, low blue). If the function's impact on the model output varies smoothly as its value changes, then this color will also have a smooth gradation. Hence, as presented in Figure 2, supply air from the data center cooling system and hot temperature in the data center environment have greater impact to accurately modeling and maintain data center ambient temperature.



Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 December 2022 doi:10.20944/preprints202212.0482.v1

Figure 2: 2.a. The average absolute value of the SHAP values for each feature globally, obtaining a bar graph of 10 topmost important features in relation to DC ambient temperature target variable. 2.b, SHAP summary plot of a XGBSVFIR method with 10 features of the time series is plotted. The higher SHAP value of a feature, the higher the best the model performance. Everyone in the data set is executed through the model and a point is created for each feature attribution value, so that each instance is displayed as a point on the line of each entity. Points are colored by the feature value for each instance and are accumulated vertically to show the density.

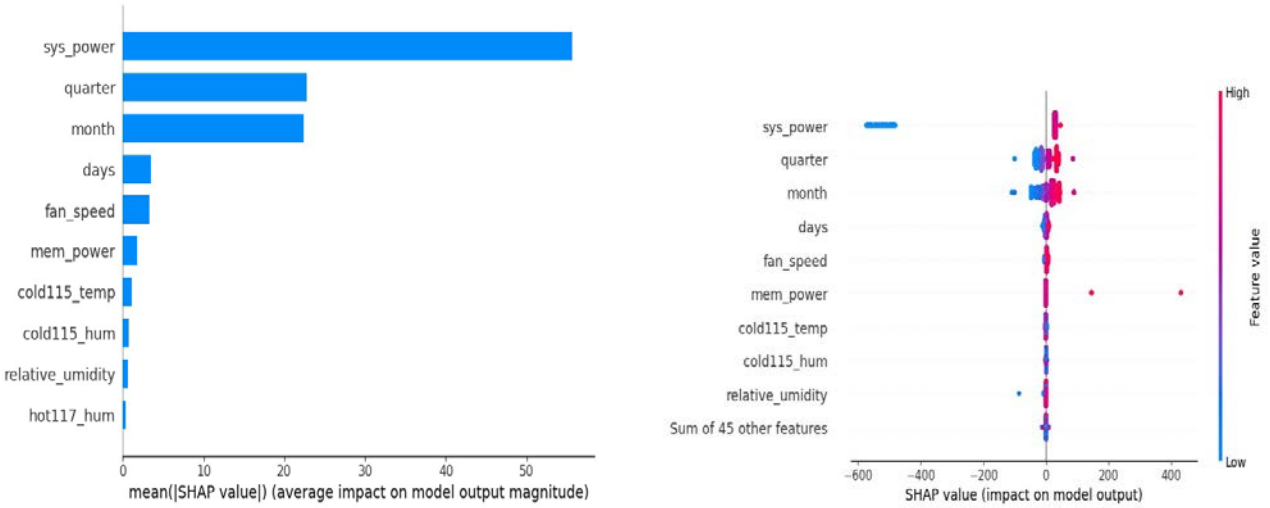


Figure 3: 3.a the average absolute value of the SHAP values for each feature globally, obtaining a bar graph of 10 topmost important features in relation to DC ambient temperature target variable. 3.b, SHAP summary plot of a XGBSVFIR method with 10 features of the time series is plotted. The higher SHAP value of a feature, the higher the best the model performance. Each individual in the data set is executed through the model and a point is created for each feature attribution value, so that each instance is displayed as a point on the line of each entity. Points are colored by the feature value for each instance and are accumulated vertically to show the density.

The partial dependence plots represent in Figures 6 and 7 demonstrates partial dependency of a single feature over the total population. The feature values vary, and the output of the model is also changed. Hence, the model outputs change as the features change helps us to explain how the model depends on that feature. An alternative to these plots using the SHAP values are the SHAP dependence plots. The impact of supply air, return _air extends over a relatively wide range of population. To understand how these features affect the model output, i.e., how the importance attributed to the features change as its values changes. We can plot the SHAP value of these features vs. the value of these feature for all the examples in a dataset, Figure 6 and 7 with respect to the target variables. While standard partial dependence plot only produces lines, SHAP values are represented as a function where each point represents an in- stance of the data set. In this way, SHAP dependence plot capture vertical dispersion due to the interaction effects in the model. The horizontal axis shows the real value of the features, the vertical axis shows the effect of each feature on the prediction and interaction effects can be visualized by coloring each point with the value of another feature. Besides, this information is shown in three axes so that we better capture the relationships between variables.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 December 2022 doi:10.20944/preprints202212.0482.v1

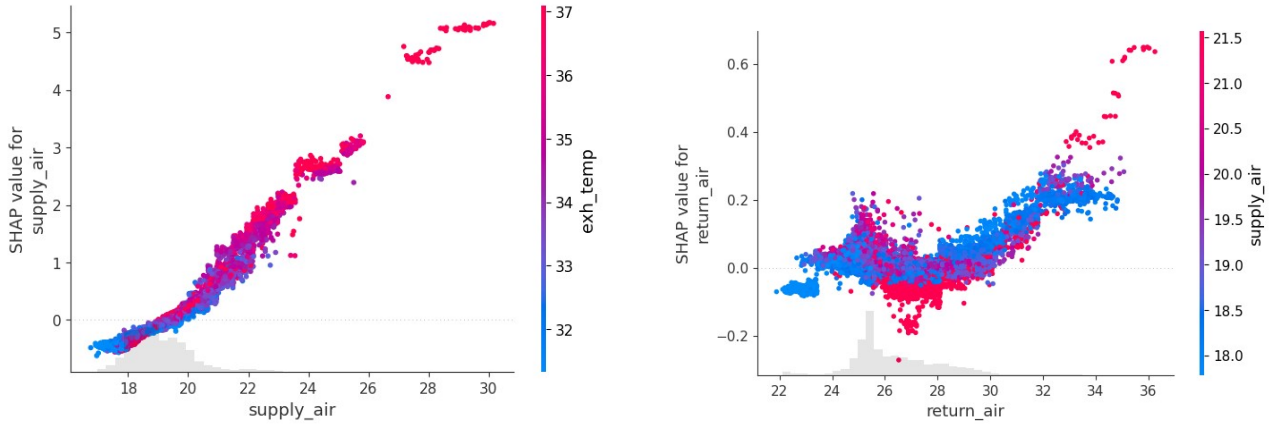


Figure 6: SHAP dependence plots of influential features of the time series samples in relation to DC ambient temperature target variable. Each point is an instance. The x-axis represents one feature, and the y-axis represents the SHAP value attributed to that feature. Each point is colored by another feature.

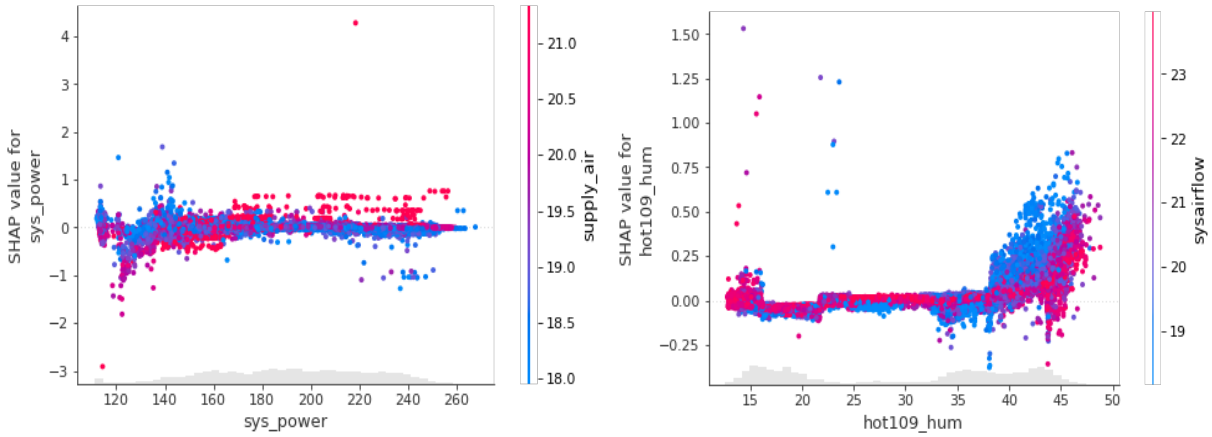


Figure 7: SHAP dependence plots of influential features of the time series samples in relation to DC ambient temperature target variable. Each point is an instance. The x-axis represents one feature, and the y-axis represents the SHAP value attributed to that feature. Each point is colored by another feature.

5 Conclusions and Future works

In conclusion, we introduced SHAP values-assisted feature subset selection (FSS) method for the identification of relevant features in multivariate time series (MTS) problems in the context of a data center. It is a class of additive feature attribution values that obey desirable accuracy, missingness, and consistency properties. It is also more consistent in attributing feature importance

 doi:10.20944/preprints202212.0482.v1

and more in line with human intuition. Furthermore, SHAP addresses the high frequency and cardinality of features that occurred in feature importance-based feature selection methods. SHAP computed the importance of each feature based on Game Theory concepts that calculate the contributions of each feature towards model development. As a result, the SHAP-value-based FSS method is useful for identifying relevant FSS to effectively modeling data center operations while providing insight into the data, improving model performance, and lowering computational expenses. Understanding the underlying patterns enables data center operators to make data-driven decisions while maintaining their data center operations ahead of time to ensure service continuity and resource availability. We proved the effectiveness of the SHAP-assisted FSS method compared with several commonly used feature selection approaches using real data streams obtained from an HPC data center (ENEA CRESCO6) cluster. We demonstrated the experiment by picking 10 of the most significant features of each method. The results in Table 2 and Table 3 demonstrated that, with better interpretability, the SHAP-assisted FSS method outperformed others commonly used feature selection methods discussed in this paper. Unlike other methods, SHAP is also a model-agnostic approach that can be applied to machine learning and deep learning techniques.

In future work, the method needs to be further investigated with more data and validated with other additive feature attribution methods. We will extend our investigation to apply the SHAP method for explaining complex black-box models and determining controllable features to find out the optimal solution for optimizing data center operations. We will also extend our work to investigate SHAP method for identifying important features in real-time predictions in both machine learning and deep learning forecasting models.

References

[1] A. Mal kowska, M. Urbaniec, and M. Kosal a, “The impact of digital transformation on European countries: Insights from a comparative analysis,” *Equilibrium. Quarterly Journal of Economics and Economic Policy*, vol. 16, no. 2, pp. 325–355, 2021.

[2] M. S. Hoosain, B. S. Paul, and S. Ramakrishna, “The impact of 4ir digital technologies and circular thinking on the United Nations sustainable development goals,” *Sustainability*, vol. 12, no. 23, p. 10143, 2020.

[3] J. Nicholson, “How is coronavirus impacting the news? our analysis of global traffic and coverage data,” *Chartbeat, March*, vol. 25, 2020.

[4] A. S. Andrae and T. Edler, “On global electricity usage of communication technology: Trends to 2030,” *Challenges*, vol. 6, no. 1, pp. 117–157, 2015.

[5] R. Bianchini, M. Fontoura, E. Cortez, *et al.*, “Toward ml-centric cloud platforms,” *Communications of the ACM*, vol. 63, no. 2, pp. 50–59, 2020.

[6] R. Evans and J. Gao, “Deepmind ai reduces google data centre cooling bill by 40%,” *DeepMind blog*, vol. 20, p. 158, 2016.

[7] A. Grishina, M. Chinnici, A.-L. Kor, E. Rondeau, and J.-P. Georges, “A machine learning solution for data center thermal characteristics analysis,” *Energies*, vol. 13, no. 17, p. 4378, 2020.

[8] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” *Advances in neural information processing systems*, vol. 30, 2017.

 doi:10.20944/preprints202212.0482.v1

[9] X. Xiaomao, Z. Xudong, and W. Yuanfang, "A comparison of feature selection methodology for solving classification problems in finance," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1284, 2019, p. 012026.

[10] N. J. Vickers, "Animal communication: When i'm calling you, will you answer too?" *Current biology*, vol. 27, no. 14, R713–R715, 2017.

[11] L. C. Molina, L. Belanche, and A. Nebot, "Feature selection algorithms: A survey and experimental` evaluation," in *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, IEEE, 2002, pp. 306–313.

[12] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[13] P. Cunningham, B. Kathirgamanathan, and S. J. Delany, "Feature selection tutorial with python examples," *arXiv preprint arXiv:2106.06437*, 2021.

[14] G. Wei, J. Zhao, Y. Feng, A. He, and J. Yu, "A novel hybrid feature selection method based on dynamic feature importance," *Applied Soft Computing*, vol. 93, p. 106337, 2020.

[15] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.

[16] K. Yang and C. Shahabi, "On the stationarity of multivariate time series for correlation-based data analysis," in *Fifth IEEE International Conference on Data Mining (ICDM'05)*, IEEE, 2005, 4–pp.

[17] E. C. Blessie and E. Karthikeyan, "Sigmis: A feature selection algorithm using correlation-based method," *Journal of Algorithms & Computational Technology*, vol. 6, no. 3, pp. 385–394, 2012.

[18] N. Rock, "Corank: A fortran-77 program to calculate and test matrices of pearson, spearman, and kendall correlation coefficients with pairwise treatment of missing values," *Computers & Geosciences*, vol. 13, no. 6, pp. 659–662, 1987.

[19] U. of Alabama at Birmingham and N. I. of Health (NIH), *Autoantibody reduction therapy in patients with idiopathic pulmonary fibrosis (art-ipf)*, 2018.

[20] A. Saikhu, A. Z. Arifin, and C. Fatichah, "Correlation and symmetrical uncertainty-based feature selection for multivariate time series classification," *International Journal of Intelligent Engineering and System*, vol. 12, no. 3, pp. 129–137, 2019.

[21] G. Doquire and M. Verleysen, "Feature selection with missing data using mutual information estimators," *Neurocomputing*, vol. 90, pp. 3–11, 2012.

[22] B. Kathirgamanathan and P. Cunningham, "Correlation based feature subset selection for multivariate time-series data," *arXiv preprint arXiv:2112.03705*, 2021.

[23] M. A. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, The University of Waikato, 1999.

[24] Z. Yang, J. Du, Y. Lin, *et al.*, "Increasing the energy efficiency of a data center based on machine learning," *Journal of Industrial Ecology*, vol. 26, no. 1, pp. 323–335, 2022.

[25] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

 doi:10.20944/preprints202212.0482.v1

[26] A. Shrikumar, P. Greenside, and A. Kundaje, “Learning important features through propagating activation differences,” in *international conference on machine learning*, PMLR, 2017, pp. 3145– 3153.

[27] T. K. Ho, “Random decision forests,” in *Proceedings of 3rd international conference on document analysis and recognition*, IEEE, vol. 1, 1995, pp. 278–282.

[28] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[29] ——, “Bagging predictors,” *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[30] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd ACM sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[31] S. M. Lundberg, G. G. Erion, and S.-I. Lee, “Consistent individualized feature attribution for tree ensembles,” *arXiv preprint arXiv:1802.03888*, 2018.