State discovery and prediction from multivariate sensor data

Olli-Pekka Rinta-Koski[1], Miki Sirola[1], Le Ngu Nguyen[1], and Jaakko Hollmén[1,2]

[1] Aalto University, Department of Computer Science, Espoo, Finland
[2] Department of Computer and Systems Sciences, Stockholm University, Stockholm, Sweden
jaakko.hollmen@aalto.fi

Abstract. The advent of cloud computing and autonomous data centers operating fully without human supervision has highlighted the need for fault-tolerant architectures and intelligent software tools for system parameter optimization. Demands on computational throughput have to be balanced with environmental concerns, such as energy consumption and waste heat. Using multivariate time series data collected from an experimental data center, we build a state model using clustering, then estimate the states represented by the clusters using both a hidden Markov model and a long-short term memory neural net. Knowledge of future states of the system can be used to solve tasks such as reduced energy consumption and optimized resource allocation in the data center.

1 Introduction

Cloud computing is the domain of connected and distributed computing resources, where the end user does not need to be concerned about the resource topology and nature. As systems grow in complexity, the need for intelligent, self-managing architectures becomes evident. This domain of computing systems that manage themselves autonomously according to goals set by human administrators is called autonomic computing [14]. An autonomous data center is able to operate independently, handling issues such as intermittent power failure, faulty components, and overheating without human intervention.

The AutoDC project [11] was started by a consortium of both industrial and academic partners to bring together an innovative design framework for autonomous data centers. To this end, we have investigated machine learning techniques and their applicability to problems in this domain. The machine learning goals of the project are as follows: "A powerful data analytics engine is required to achieve data collection from the various monitoring systems, which is then consolidated with external data sources and periodically stored as appropriate records to allow for both real-time and off-line ecosystem modelling and machine learning data analysis. The analytics results will ensure proper actions are applied to the control systems for optimised power, cooling, network and server operation, which is essential to maintain the data center 'health' within desired parameters to reach identified target key performance indicator (KPI) values."

In addition to the data flowing through and being processed in the data center, there is metadata being generated that has to do with the operational state of the data center itself. The amount of sensor data collected at the data center is huge, making manual annotation difficult or impossible. Our approach is to use unsupervised learning methods to organize the sensor data into states of the system, then build a model for prediction of these states. The purpose of the model is to provide an autonomous method for adjusting the control parameters of the data center, with the hope of achieving better performance in terms of resource use. Possible optimization goals include thermal efficiency and CPU utilization.

The rest of the paper is organized as follows: Section 2 outlines our proposed approach. Section 3 presents the clustering and prediction results from our experiments. Finally, we summarize our findings and conclude the paper in Section 4.

1.1 Prior work

Previous work in this domain includes system state discovery using clustering of system log data [18] and resource usage data [8, 7], novelty detection in projected spaces [25], using state change detection for history-based failure prediction [20], and forecasting for decision making support in autonomous systems [3].

Time series analysis in the data center has been applied to a number of interesting problems such as predictive maintenance, traffic balancing, and anomaly detection. Ahuja et al. [1] have used supervised learning to predict data center power variation. They have used a support vector machine [4] approach to predict power variation 15 minutes into the future, in order to have a reasonable time period for applying changes to control parameters. Aibin et al. [2] have used Monte Carlo tree search [16] for prediction of traffic between data centers. Shi et al. [23] have used a number of machine learning models, including long-short term memory [9], for anomaly detection from data center activity logs. Yang et al. [26] used regression methods for hardware failure prediction.

2 Methods

2.1 Summary

Our approach can be summarized as follows. First, multivariate time series data is preprocessed to reduce dimensionality. Then, system states are identified by clustering these low-dimension projections. Finally, the clustering results are used to build a model for state transition prediction. Two different approaches, hidden Markov model (HMM) [19] and long short-term memory (LSTM) [9], are discussed. Figure 1 shows a high-level view of the process for analyzing sensor data collected from a data center.

We have studied a multivariate time series data set obtained from the EDGE small data center testbed at the RISE ICE Datacenter in northern Sweden [5].

This experimental data set was created by creating a number of diverse load patterns for the purpose of cooling system performance evaluation under varying conditions. The data consists of timestamped sensor observations. Computational values include server CPU and core load. Environmental sensors include humidity and temperature.

Our model tries to identify system states based on the projection of the data onto a low-dimensional space defined by the first c principal components. Selection of c will be discussed later. These vectors are then clustered with k-means clustering [17]. These clusters are taken to represent system states. Predicting future system states makes it possible to anticipate system load and tune control parameters according to criteria such as throughput or heat emission reduction.

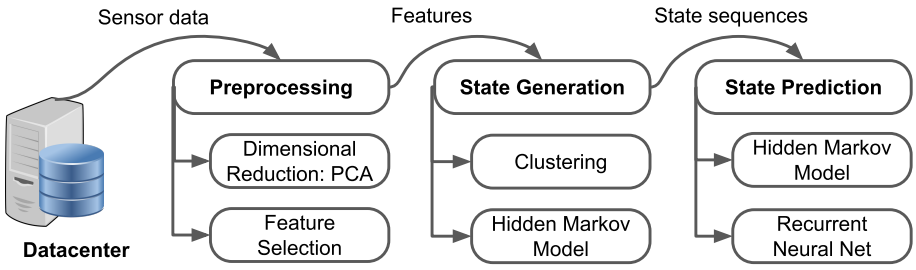

Fig. 1. A general process to analyze sensor data collected from a data center.

2.2 Data set

Our data set contains 5072 timestamped measurements of 44 variables sampled from sensors placed in an autonomous datacenter over a time period of slightly over 42 hours. The measured values include features such as power, CPU temperature, fan signal, fan power, chamber temperature, and ambient temperature. Each variable has been standardized by removing the mean and scaling to unit variance, missing values have been imputed using the mean of each variable, and the data has been resampled to uniform 30s intervals.

2.3 Principal component analysis

Principal component analysis (PCA) [12] is a classical statistical technique for linear dimension reduction of multivariate data. In PCA, the goal is to represent the original d-dimensional data with a new orthogonal coordinate system that has d base vectors. By choosing a lower dimensionality starting with the most dominant base vectors, called the principal components, we can achieve a dimensionality reduction of the original data by selecting a subset of c eigenvectors. The PCA is solved by solving a corresponding eigenvalue problem. In

the original data matrix $\mathbf{X} = (x_{ij}), I = 1, \ldots, n, j = 1, \ldots, d$, the entry x_{ij} denotes jth component of the ith data entry. The projection matrix $\mathbf{A}$ consisting of eigenvectors can be used to project the original data into a new coordinate system as $y = \mathbf{A}x$.

We are interested in lowering the dimensionality of the high-dimensional sensor data in order to visualise the data in lower dimensions, and to use the lower dimensional representation as a starting point for further analysis. The phenomenon called the curse of dimensionality [24] tells us that higher dimensional data spaces are more difficult to model with finite data. Our point of investigation is to see whether the lower dimensional representation is a beneficial starting point for solving subsequent tasks.

In this paper, our goal is to extract and to describe operational states of a system, where a high dimensional sensor data describes the operation. In order to define and describe states of the system, we apply clustering to the data. In order to be beneficial, the clustering solution based on the lower dimensional data should be somehow superior to the solution generated from the original high dimensional data.

2.4 Clustering with the k-means algorithm

In k-means clustering [17], clusters are represented as prototypes which are local averages of the data closest to the cluster. Data points are first assigned to clusters, and the cluster memberships are iteratively adjusted according to distances to neighbouring data points.

We apply the following procedure for the analysis: We have a high-dimensional data set ($d = 44$) describing the operation of a rack in a data center and its operation environment in terms of temperature and humidity. We lower the dimensionality of the original data d to d'. Then we attempt to solve the clustering problem for k clusters ($k \in \{2, \ldots, 19\}$) using the k-means algorithm. We measure the goodness of clustering by two measures, the Davies-Bouldin index [6] and the Silhouette score [21]. The presented results are based on 5-fold cross-validation repeated 5 times.

2.5 Dynamic modeling with hidden Markov model

We have used HMM [19] for estimating system state. HMM is a good fit for the problem, since the actual system state is not known but things about it can be inferred from sensor readings. For the HMM implementation, sensor outputs stand for emission observations, system states (HMM hidden states) are represented by the clusters, and transition probabilities are what we aim to estimate in order to model the dynamic behaviour of the system.

Bayesian Information Criterion (BIC) [22], which is an estimator of prediction error, was used for selecting an appropriate number of HMM states. BIC maximizes the Bayesian posterior probability.

2.6 State prediction with long short-term memory

Recurrent neural networks are used to model the dependency of patterns in data. They have two issues: vanishing gradient and exploding gradient [10]. LSTM [9] models were introduced to resolve these issues by integrating a memory cell. The cell can capture dependencies of data over arbitrary time periods and its three gates regulate the information flow. Unlike HMMs where the Markov assumption means that the past does not affect transition probabilities, LSTMs can model arbitrary temporal dependencies.

In this work, the input of our LSTM model is a sequence of multivariate sensor data (e.g. temperature, humidity, server load, fan speed, ...). The model output is the state of the data center, which is reflected by the sensor data. Since we relied on unsupervised learning methods to analyze the data, the labels (i.e. state sequences) used to train the LSTM model were generated by the kmeans clustering algorithm and the HMM. After obtaining a sequence of states, it is possible to use a LSTM model to predict future states.

3 Results

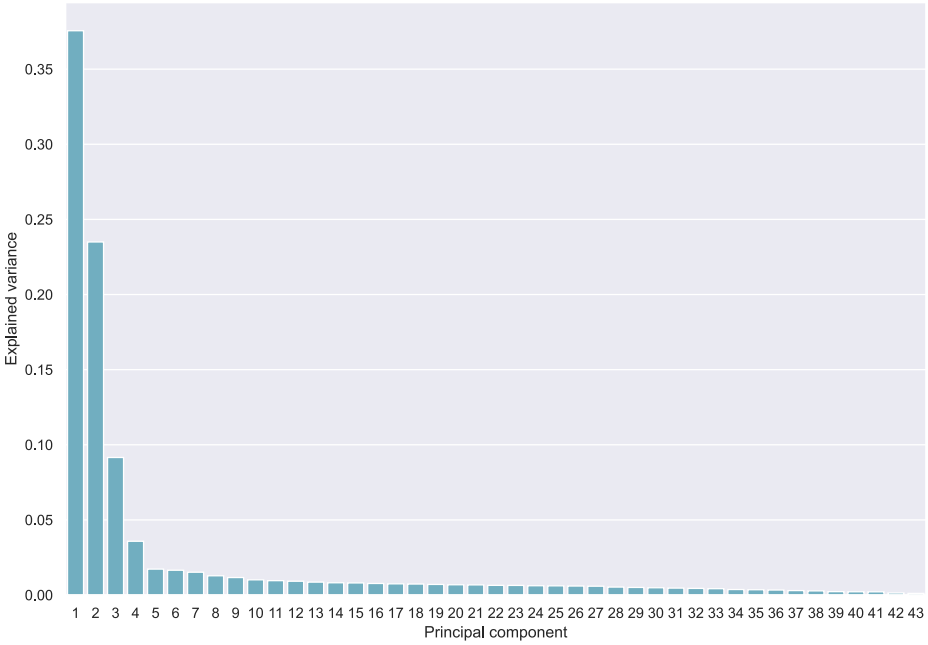

Fig. 2. Principal component scree plot.

3.1 Dimensionality reduction with principal component analysis

After using PCA to reduce dimensionality, Figure 2 shows the explained variance of each of the principal components. It can be seen that the knee of the scree plot is at or near the fourth component. Using the Kaiser criterion [13], we retained the first four components.

3.2 Clustering with k-means

The original data vectors of dimension 44 are replaced with vectors consisting of the first four principal components only. These are then grouped using k-means clustering. (Figure 3 shows the data grouped into 3 clusters using the first 3 principal components.) These clusters are taken to be the states of the system. With very limited information available of the physical process, interpretation of the states is difficult and challenging. By looking at the PCA loadings in our analysis we get information about which variables are dominating in each of the PCA components, which opens opportunities to interpret the connections between states and measured variable behaviour.

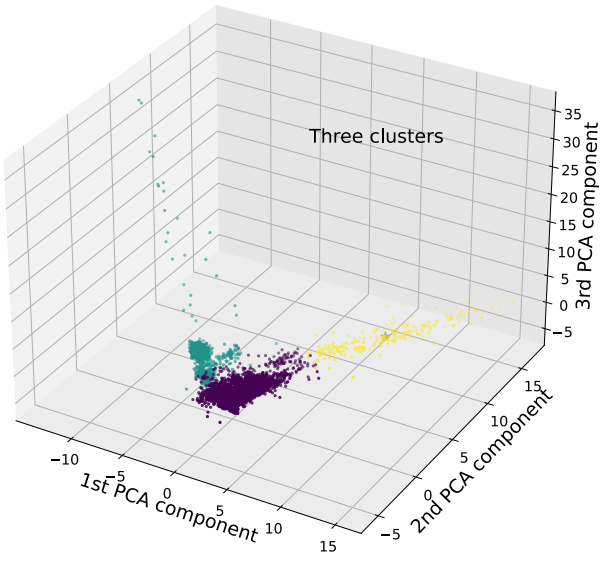

Fig. 3. Data division into three states by clustering.

By varying the number of clusters and computing the corresponding Davies-Bouldin indexes and Silhouette scores, we have found that for this particular data set, $k = 3$ achieves the best separation (Figure 4).

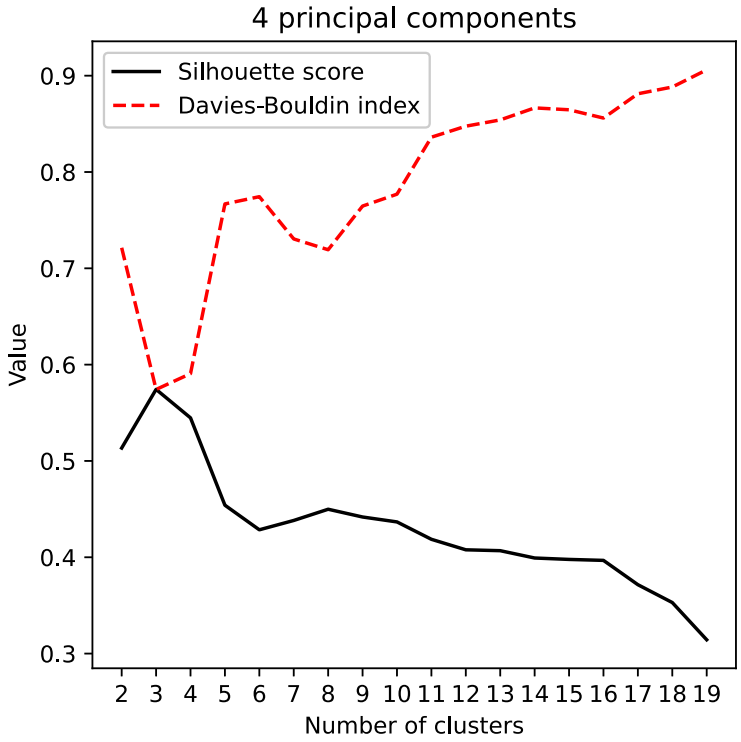

Fig. 4. Davies-Bouldin indexes and Silhouette scores for $k \in \{2, \ldots, 19\}$ clusters of vectors of 4 principal components. Lower values of Davies-Bouldin index and higher values of Silhouette score indicate better clustering.

It is possible to notice some obvious correlations between some variables and variable groups. Typically power, temperature, utilization and fan speed are associated with each other, while humidity has an inverse correlation. The effect of some variables may be delayed. Room temperature control settings may also have an effect on some variables as well as on the waste heat production of the data center.

PCA loadings tell also the variance stored in each PCA component. In our data the first four PCA components have respectively 37.1%, 23.3%, 8.7% and 2.8% of the total variance, collectively adding up to 72.0%. Table 1 shows the variables that have the largest effect on each PCA component. The scale in the left column is from 0 (no effect) to 1 (full domination). The numbers in the variables refer to the six channels. In CPU temperatures the second number refers to one of the two different cores in each channel.

Table 1. Dominating variables in each PCA component. p_{Sn} = power, y_{Smn} = CPU temperature, u_{Fn} = fan signal, n_{Fn} = number of cores, p_{Fn} = fan power, x_{Sn} = load, T_c = chamber temperature, T_a = ambient temperature, * = inverse correlation.

Dominance	1st component	2nd component	3rd component	4th component
> 0.4			x_{S5}, x_{S6}, x_{S3}	T_c^*, T_a^*
> 0.35			x_{S1}, x_{S2}, x_{S4}	
> 0.3				
> 0.2		p_{F3}, T_c^*		
> 0.18	p_{S6}, p_{S1}, n_{F6}, p_{S3}, u_{F3}, n_{F2}, p_{S4}	p_{F1}, u_{F3}, p_{F4}, p_{F2}, y_{S21}^*, y_{S61}^*		
> 0.15	n_{F3}, y_{S21}, n_{F4}, p_{S2}, p_{S5}, n_{F1}, n_{F5}, u_{F5}, y_{S31}, u_{F6}, y_{S32}, y_{S61}, u_{F2}, y_{S11}, y_{S12}, u_{F4}, u_{F1}, y_{S51}, y_{S22}	u_{F5}, p_{F5}, n_{F4}, n_{F1}, n_{F2}, n_{F6}, n_{F3}, n_{F5}, u_{F1}, u_{F4}, u_{F6}, y_{S52}^*, y_{S41}^*, y_{S42}^*, y_{S12}^*, y_{S11}^*, y_{S31}^*, y_{S22}^*, y_{S62}^*, y_{S32}^*		

From Table 1 and variable plots, we can make the following observations on states and state transitions. It seems that power and CPU load (together with some correlating variables) is a dominating factor of the 1st and 2nd PCA component. The fan signal (correlating with fan power, fan speed, etc.) is another major factor in the 1st and 2nd PCA component and a reason for the smaller and bigger variations in both components. Note that the effect on the 2nd PCA component is often reversed. Variables related to utilization are strongly related to the 3rd PCA component. These variables include high peaks, see Figure 5. Some delayed temperature and humidity changes have a small effect on the 2nd and 3rd PCA component. Temperature dominates the 4th PCA component.

The 1st and 2nd PCA components define the biggest states "idle" and "CPU power". Inside the idle state there are high peaks in the 1st and 3rd PCA components. These peaks seem to appear when we move from "idle" state to "CPU power" state. The strong vibration in fan signal especially seen in the 2nd PCA component is responsible for the third state. This effect can be noticed towards the end of the time series, see Figure 5. Delayed temperature changes map onto different locations inside the "CPU power" state in the 3-D presentation. These effects within the state represent "cooler power on" and "hotter power on", the latter of which is more stable. The stability of CPU load also has an effect here, maybe because the average load is higher in stable load than in vibrating load. The interpretation of states affected by the 4th PCA component is made harder by the lack of graphical representation. The 4th component is mainly affected by the two temperature measurements (chamber and ambient) in an inverse relationship.

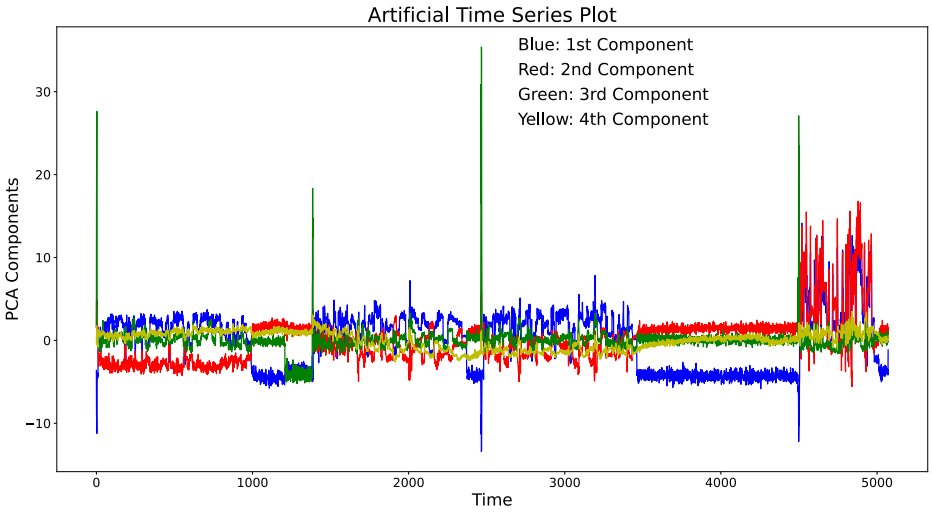

Fig. 5. PCA result in time series form.

We could label the three states as idle, high CPU load, and strongly vibrating fan signal (varying CPU load). This third state appears towards the end of the time series where the fan signal is strongly vibrating. This state is the most different to all the other states and could even be a failure state. This kind of example study demonstrates that our approach constructing states according to the physical behaviour of the process may also reveal failure states, and could be used for anomaly detection.

The described state behaviour is valid only to this type of data having a certain set of measured variables. There seem to be different state characteristics in different types of data. As an example of single measured variable types, we have experimented with large amounts of pure temperature data also collected from an experimental data center. With this data it is also possible to find similar states, but the characteristics are very different. Temperature data typically have very clear and separable clusters and states. There are also fewer characteristics such as variable delays, and the states are formed mostly by different power levels. In addition, sometimes the vibration effect forms two alternating well-separated states within one power level.

3.3 Modeling dynamic behaviour with HMM

HMM is used to model dynamic behaviour of the system. The state transition parameters model the changes in the system as one stable state turns into another. With HMM, the classification to a desired number of states is similar to k-means clustering. We have noted that with this data, HMM is more likely to minimize the state transitions and gives somewhat fewer transitions. The states and state transitions with both methods are seen in Figure 6.

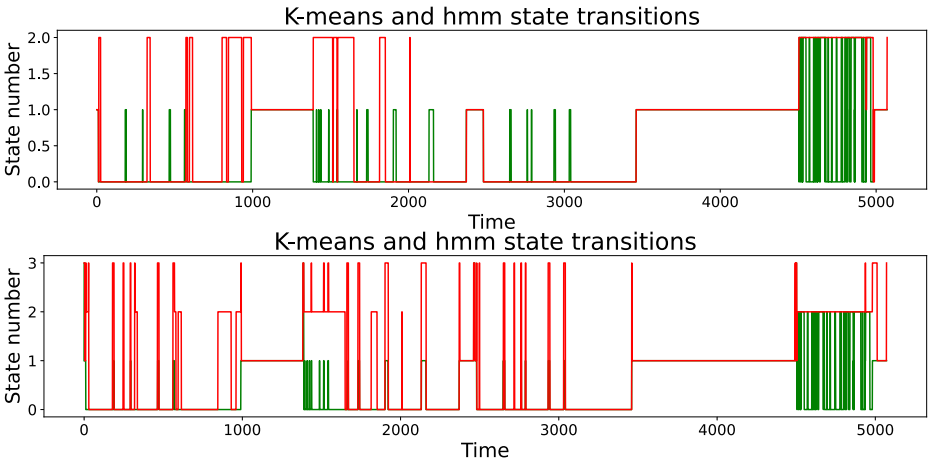

Fig. 6. States and state transitions with k-means clustering and HMM. k-means in green, HMM in red. Top: Three states. Bottom: Four states.

Prediction for state probabilities for each state is calculated, see Figure 7. Mostly one state is on (probability 1) and the other states are off (probability 0). Near the state transitions the probabilities may have also other values. One of the states only appears near the end of the time series when the fan signal is vibrating strongly.

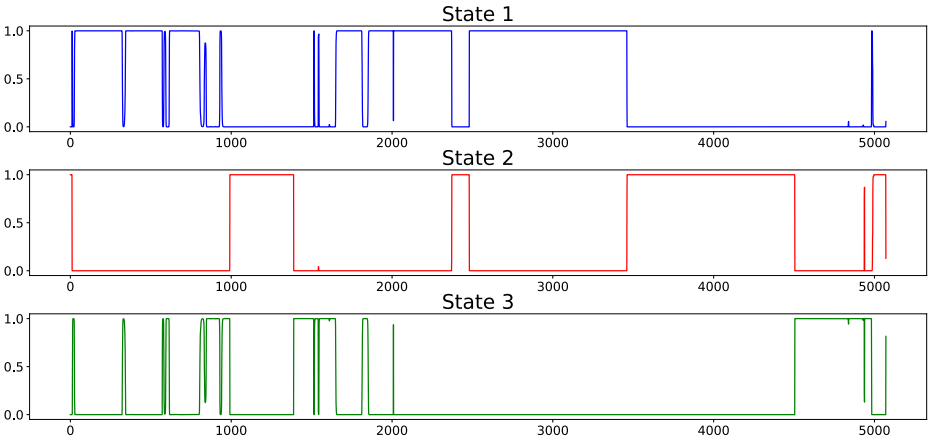

Fig. 7. State probabilities by state.

We calculated BIC according to two parameters: the number of states and the number of dimensions of the input data. We assumed that the sensor data

reflected $1 \ldots 20$ states of the data center and that dimensionality could be reduced while retaining enough of the relevant information. Figure 8 summarizes BIC with various numbers of states and dimensions. Our judgement was that $s \in \{3, 4, 5\}$ results in a good BIC value. This result agreed with the clustering metrics shown in Figure 4.

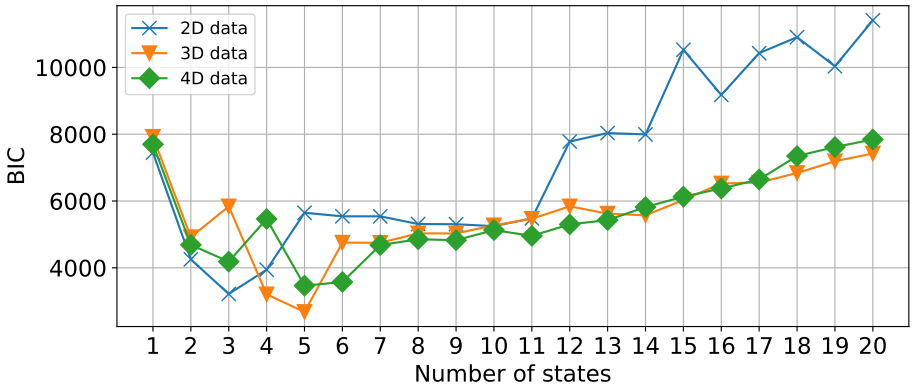

Fig. 8. Bayesian information criterion (BIC) values when varying the number of states and the number of dimensions (a model with a lower BIC is preferable).

3.4 LSTM-based state prediction

We performed five-fold cross-validation to obtain the aggregated results (i.e. averaging the accuracy and the confusion matrices). The LSTM model, built using the PyTorch library, had a hidden layer of dimension 20 and a time-step of 10 sequences. We train the LSTM model for 100 epochs using the Adam optimizer [15]. In each cross-validation round, the dataset was split into two parts: 80% of the samples for training and the remaining samples for testing.

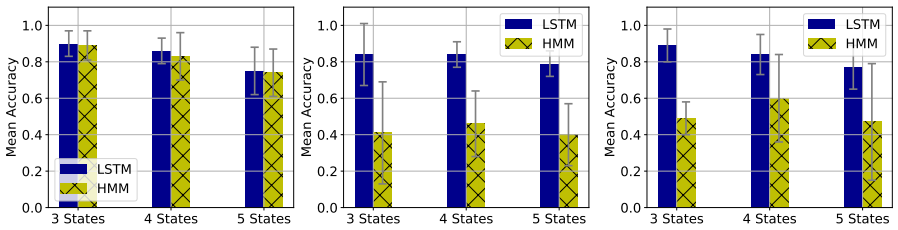

Fig. 9. Mean accuracy of hidden Markov model and long short-term memory predictions. Left: Original data of dimension 44. Middle: 3D data. Right: 4D data.

We performed the experiments with the number of states ranging from three to five. The mean accuracy of the LSTM predictions was compared to that of the HMM trained on the same dataset (also with cross-validation). The labeling of the state sequences was generated by k-means clustering. We observed that the mean accuracy of both models was similar when trained on the original data of dimension 44 (see Figure 9, left subplot). However, when using PCA to transform the data to a lower-dimensional space (i.e. 3D in Figure 9, middle subplot and 4D in Figure 9, right subplot), the mean accuracy of HMM decreased significantly compared to that of the LSTM model. This could be explained by the temporal dependency represented by the latter.

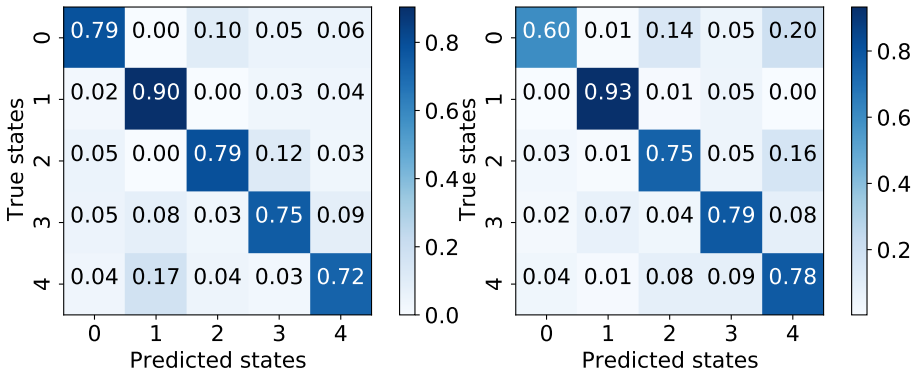

Fig. 10. Confusion matrix of LSTM predicting five states generated by clustering. Left: 3D data. Right: 4D data.

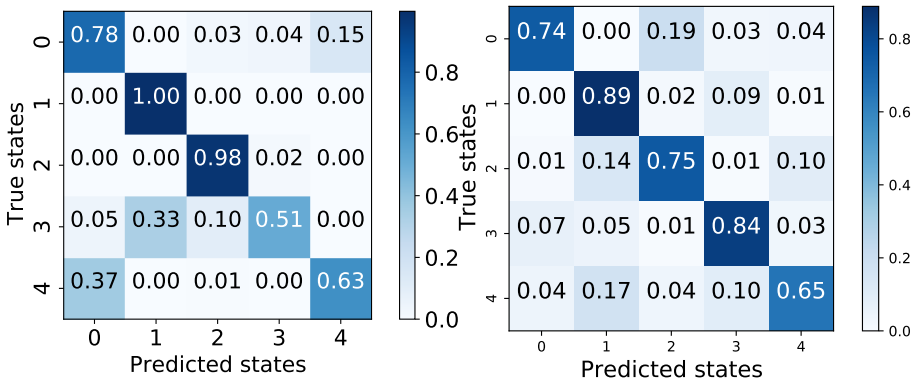

Fig. 11. Confusion matrix of LSTM predicting five states generated by HMM. Left: 3D data. Right: 4D data.

Furthermore, we investigated the predictions output by the LSTM model trained with state sequences generated by the k-means clustering algorithm (Figure 10) and the HMM (Figure 11). As discussed in 3.3, we decided to limit our interest to cases with the number of states $s \in \{3, 4, 5\}$. Out of these, we selected the five-state case as a representative example in which the multivariate sensor data was transformed into a 3D and 4D space using PCA. The confusion matrices showed that our LSTM model could predict the states of the data center using sequences of sensor data embedded in a lower dimensional space. We concluded that both methods (k-means and HMM) could be employed to generate labels for training state prediction models.

4 Summary and Conclusions

In this paper, we have investigated using multivariate time series data to build a model of operational states and state transitions in a data center. The data consisted of sensor output collected from sensors in the computational environment. PCA was used to reduce the dimensionality of our original data from 44 to 4, and the resulting lower-dimension data vectors were clustered using k-means to get a labeling of system states. This labeling was then used with both HMM and LSTM, building two different estimates of these states.

Our results show that unsupervised learning methods can help to discover states from multivariate sensor data. We applied k-means clustering and HMM to explore the possible states based on sensor data and achieved consistent results. We have identified a set of states of the data center, which we describe with a dynamic model. We are able to predict future states using supervised learning techniques. The use of dimension reduction techniques resulted in lowered computational complexity, which is especially useful when available resources are scarce.

Further work in this domain could involve expert interpretations of the state discovery to define how the states map onto real operational states. These results could then be applied to control data center parameters. Replicating these results with data from other data centers would make these results more generally applicable. This would require careful examination of variable selection, since the set of available sensors might be very different. Comparison with other projection techniques, such as randomized projections, remains an interesting avenue for the future.

State discovery by itself can be used to implement data visualization for monitoring. A prediction of the operational state of the data center could be used to adjust system parameters such as cooling system power or CPU allocation. Different concurrent goals such as lower power use and increased throughput may be in conflict, so a policy to optimize how the system works could be guided by intelligent algorithms that make predictions on how demands on the system will change in the short term.

Acknowledgments We acknowledge the computational resources provided by the Aalto Science-IT project. We thank Rickard Brännvall and Jonas Gustafsson of RISE ICE Datacenter for their help with the data set.

References

[1] Nishi Ahuja et al. "Power Variation Trend Prediction in Modern Datacenter". In: *2017 16th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)*. 2017 16th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm). Orlando, FL: IEEE, May 2017, pp. 977–980.

[2] Michał Aibin et al. "Traffic Prediction for Inter-Data Center Cross-Stratum Optimization Problems". In: *Proceedings*. 2018 International Conference on Computing, Networking and Communications (ICNC): Optical and Grid Computing. 2018, p. 6.

[3] Andre Bauer et al. "Time Series Forecasting for Self-Aware Systems". In: *Proceedings of the IEEE* 108.7 (July 2020), pp. 1068–1093.

[4] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. "A Training Algorithm for Optimal Margin Classifiers". In: *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*. ACM Press, 1992, pp. 144–152.

[5] Rickard Brännvall et al. "EDGE: Microgrid Data Center with Mixed Energy Storage". In: *Proceedings of the Eleventh ACM International Conference on Future Energy Systems*. E-Energy '20: The Eleventh ACM International Conference on Future Energy Systems. Virtual Event Australia: ACM, June 12, 2020, pp. 466–473.

[6] David L. Davies and Donald W. Bouldin. "A Cluster Separation Measure". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-1.2 (Apr. 1979), pp. 224–227.

[7] Nentawe Gurumdimma and Arshad Jhumka. "Detection of Recovery Patterns in Cluster Systems Using Resource Usage Data". In: *2017 IEEE 22nd Pacific Rim International Symposium on Dependable Computing (PRDC)*. 2017 IEEE 22nd Pacific Rim International Symposium on Dependable Computing (PRDC). Christchurch, New Zealand: IEEE, Jan. 2017, pp. 58–67.

[8] Nentawe Gurumdimma et al. "CRUDE: Combining Resource Usage Data and Error Logs for Accurate Error Detection in Large-Scale Distributed Systems". In: *2016 IEEE 35th Symposium on Reliable Distributed Systems (SRDS)*. 2016 IEEE 35th Symposium on Reliable Distributed Systems (SRDS). Budapest, Hungary: IEEE, Sept. 2016, pp. 51–60.

[9] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (Nov. 1, 1997), pp. 1735–1780.

[10] Sepp Hochreiter et al. “Gradient Flow in Recurrent Nets: The Difficulty of Learning Long-Term Dependencies”. In: *A Field Guide to Dynamical Recurrent Networks*. Ed. by John F. Kolen and Stefan C. Kremer. IEEE, 2001.

[11] ITEA3/AutoDC. *About AutoDC*. URL: `https://autodc.tech/about/` (visited on 08/08/2021).

[12] Ian T. Jolliffe. *Principal Component Analysis*. 2nd ed. Springer, 2002.

[13] Henry F. Kaiser. “The Application of Electronic Computers to Factor Analysis”. In: *Educational and Psychological Measurement* 20.1 (Apr. 1960), pp. 141–151.

[14] J.O. Kephart and D.M. Chess. “The Vision of Autonomic Computing”. In: *Computer* 36.1 (Jan. 2003), pp. 41–50.

[15] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: 3rd International Conference on Learning Representations (ICLR). San Diego, CA, USA, 2015. arXiv: `1412.6980`.

[16] Levente Kocsis and Csaba Szepesvári. “Bandit Based Monte-Carlo Planning”. In: *Machine Learning: ECML 2006*. Ed. by Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou. Vol. 4212. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 282–293.

[17] S. Lloyd. “Least Squares Quantization in PCM”. In: *IEEE Transactions on Information Theory* 28.2 (Mar. 1982), pp. 129–137.

[18] Adetokunbo Makanju, A. Nur Zincir-Heywood, and Evangelos E. Milios. “System State Discovery Via Information Content Clustering of System Logs”. In: *2011 Sixth International Conference on Availability, Reliability and Security*. 2011 Sixth International Conference on Availability, Reliability and Security (ARES). Vienna, Austria: IEEE, Aug. 2011, pp. 301–306.

[19] L.R. Rabiner. “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition”. In: *Proceedings of the IEEE* 77.2 (Feb. 1989), pp. 257–286.

[20] Raghunath Rajachandrasekar, Xavier Besseron, and Dhabaleswar K. Panda. “Monitoring and Predicting Hardware Failures in HPC Clusters with FTB-IPMI”. In: *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum*. 2012 26th IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). Shanghai, China: IEEE, May 2012, pp. 1136–1143.

[21] Peter J. Rousseeuw. “Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis”. In: *Journal of Computational and Applied Mathematics* 20 (Nov. 1987), pp. 53–65.

[22] Gideon Schwarz. “Estimating the Dimension of a Model”. In: *The Annals of Statistics* 6.2 (1978), pp. 461–464.

[23] Jia Shi, Gang He, and Xinwen Liu. “Anomaly Detection for Key Performance Indicators Through Machine Learning”. In: *2018 International Conference on Network Infrastructure and Digital Content (IC-NIDC)*.

2018 International Conference on Network Infrastructure and Digital Content (IC-NIDC). Guiyang: IEEE, Aug. 2018, pp. 1–5.

[24] Michael Steinbach, Levent Ertöz, and Vipin Kumar. "The Challenges of Clustering High Dimensional Data". In: *New Directions in Statistical Physics: Econophysics, Bioinformatics, and Pattern Recognition*. Ed. by Luc T. Wille. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 273–309.

[25] Janne Toivola, Miguel A. Prada, and Jaakko Hollmén. "Novelty Detection in Projected Spaces for Structural Health Monitoring". In: *Advances in Intelligent Data Analysis IX*. Ed. by Paul R. Cohen, Niall M. Adams, and Michael R. Berthold. Vol. 6065. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 208–219.

[26] Wenjun Yang et al. "Hard Drive Failure Prediction Using Big Data". In: *2015 IEEE 34th Symposium on Reliable Distributed Systems Workshop (SRDSW)*. 2015 IEEE 34th Symposium on Reliable Distributed Systems Workshop (SRDSW). Montreal, QC, Canada: IEEE, Sept. 2015, pp. 13–18.