Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 9058048, 7 pages
https://doi.org/10.1155/2022/9058048

Research Article

Learning the Correlations between IoT Systems Consisting of Massive Sensors

Shuze Jia, You Ma, Juan Xue, and Aijun Zhu

National Satellite Meteorological Center, Beijing 100081, China

Correspondence should be addressed to You Ma; mayou0531@126.com

Received 31 March 2022; Accepted 7 May 2022; Published 24 May 2022

Academic Editor: Yan Huo

An IoT system often consists of many sensors to collect data in different aspects. Meanwhile, all these sensors describe the IoT system's functional status, to which it belongs. The correlations between subsystems are always emphasized for a complex system that contains several IoT subsystems. At the same time, there are still no good ways to calculate these types of correlations since that (1) multiple sensors describe an IoT system as a matrix while the correlation between matrices cannot be calculated by the traditional methods (i.e., vector ways such as Pearson correlation coefficient) and (2) AI methods such as neural networks were introduced to resolve this problem; however, these black-box approaches cannot explain the mathematical mechanisms, and lots of memory or time are consumed. This paper proposed a novel approach named the matrix-oriented correlation computing method (MOCC) to learn the correlations between IoT systems. The critical problem of this proposed method is calculating the correlation between two curved surfaces, which are modeled as matrices, since an IoT system often contains many sensors which characterize different aspects of this system and continuously generate data in time series. By our MOCC method, the correlation or interaction between any two subsystems can be accurately measured, which means that we can predict the state of a system by its most important related system. Missing data value prediction based on our MOCC method is also presented in this paper. We verified the efficiency and effect of our proposed method via a satellite, a typical IoT system consisting of massive sensors, and the experimental result was proved to outperform existing methods.

1. Introduction

A complex system often contains several IoT subsystems. For example, a satellite platform consists of at least energy, attitude, propulsion, thermal control, data transmission, and payload six subsystems; and each subsystem is designed as the integration of smaller subsystems. The correlation between two subsystems is always an essential consideration in the maintenance or analysis of a complex IoT system [1]. By the correlation analysis, we can determine the most relevant factor of an IoT subsystem. For example, we can find that a system's data anomaly will affect other systems' functional status if there are high correlations between them [2]; or a system's data missing can be predicted based on its high relevant systems [3–7]. Generally, this type of correlation can only be calculated by the sensor data of IoT systems since the sensors are designed for monitoring data for all different properties. If a subsystem consists of M sensors that have collected data in a time series of length N, then the data would be modeled as an $M \times N$ matrix, and the correlation between two systems is actually the correlation between two matrices.

There are many existing correlation computing methods, such as Pearson correlation coefficient (PCC) [8] and cosine coefficient (COS) [9], and new correlation measuring methods have been presented in recent years [10–12]. However, most of the existing methods can only measure the correlation between one-dimensional vectors but not matrices. Figure 1 compares the vector and matrix oriented similarity measurement.

AI methods such as neural networks were introduced to resolve this problem in recent years [13]; however, these black-box approaches cannot explain the mathematical mechanisms, and lots of memory or time are consumed.

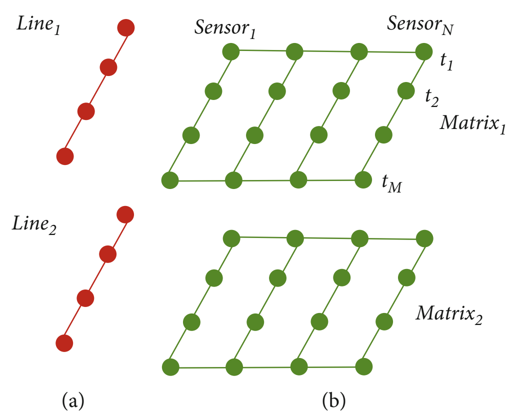

Figure 1: (a) shows that the traditional method can only measure the correlation between two vectors, meaning that each vector consists of only one sensor's data, while (b) shows two matrices, each of them consisting of several sensors.

This paper proposed a novel approach named the matrix oriented correlation computing method (MOCC) to learn the correlations between IoT systems, and this method can also be used for any multidimensional observation objects. After the M sensors of an IoT system have produced data in a time series of length N, we can get an $M \times N$ matrix. Since this matrix is constructed along with time, the MOCC method should measure the correlation between two matrices considering the time factor.

The rest of this paper is organized as follows. Section 2 presents our MOCC method. Section 3 describes a critical application—missing data prediction of our MOCC method. Section 4 describes our experiments. Section 5 concludes the paper, and Section 6 shows related work.

2. Proposed MOCC Method

For ease of explanation, we use the following matrix shown in Table 1 to describe the data produced by M sensors in a length N time series.

This section will illustrate the mathematical steps of our MOCC method to compute the correlation between any two matrices of this type. MOCC is inspired by a significant and novel math concept identified as distance correlation. The concept of distance correlation extends the correlation calculation from one-dimensional space to two-dimensional space, but it does not consider the time factor. This section presents the mathematical principles and advantages of distance correlation and then improves it by bringing the time factor into it, which eventually resulted in our MOCC method.

2.1. Distance Correlation

2.1.1. Mathematical Principles. We use the distance correlation concept to measure correlation considering multiple sensors of an IoT system integrally. Distance correlation—a statistics and probability theory-based concept—was proposed by Szekely et al. to measure the statistical dependence between two random vectors of arbitrary, not necessarily equal dimension [14, 15]. Therefore, distance correlation can measure the correlation between any two matrices more accurately and comprehensively. The distance correlation is defined as follows.

Table 1: Data of an IoT system.

	T_1	T_2	…	T_N
Sensor_1	v_{11}	v_{12}	…	v_{1N}
Sensor_2	v_{21}	v_{22}	…	v_{2N}
…	…	…	…	…
Sensor_M	v_{M1}	v_{M2}	…	v_{MN}

Let X and Y denote two different IoT systems consisting of p and q sensors, respectively. If these two systems continuously produce data in a time series of length N, then, we can get an observed random sample from the joint distribution of random vectors X in $\mathbb{R}^p$ and Y in $\mathbb{R}^q$ as follows:

$$(X, Y) = \{(X_n, Y_n): n = 1, 2, \cdots, N\}. \quad (1)$$

For example, if X is an IoT system shown in Table 1, then X_n is the n-th column of Table 1.

From the definition of X and Y, we can get that two systems that need to measure correlation need not have the same number of sensors but have to be observed in the same time series.

And define:

$$a_{kl} = \|X_k - X_l\|_p, \quad (2)$$

$$\|X\|_p = \left(\sum_{i=1}^{p} |x_i|^p\right)^{1/p}, \quad (3)$$

$$\bar{a}_{k\cdot} = \frac{1}{N}\sum_{l=1}^{N} a_{kl}, \quad (4)$$

$$\bar{a}_{\cdot l} = \frac{1}{N}\sum_{k=1}^{N} a_{kl}, \quad (5)$$

$$\bar{a}_{\cdot\cdot} = \frac{1}{N^2}\sum_{k,l=1}^{N} a_{kl}, \quad (6)$$

$$A_{kl} = a_{kl} - \bar{a}_{k\cdot} - \bar{a}_{\cdot l} + \bar{a}_{\cdot\cdot}. \quad (7)$$

Similarly,

$$b_{kl} = \|Y_k - Y_l\|_q, \quad (8)$$

$$\|Y\|_q = \left(\sum_{i=1}^{q} |y_i|^q\right)^{1/q}, \quad (9)$$

$$\bar{b}_{k\cdot} = \frac{1}{N}\sum_{l=1}^{N} b_{kl}, \quad (10)$$

$$\bar{b}_{\cdot l} = \frac{1}{N}\sum_{k=1}^{N} b_{kl}, \quad (11)$$

$$\bar{b}_{\cdot\cdot} = \frac{1}{N^2}\sum_{k,l=1}^{N} b_{kl}, \quad (12)$$

$$B_{kl} = b_{kl} - \bar{b}_{k\cdot} - \bar{b}_{\cdot l} + \bar{b}_{\cdot\cdot}. \quad (13)$$

Before giving the distance correlation between X and Y, their distance variance is defined as follows.

$$\nu^2(X, Y) = \frac{1}{N^2}\sum_{k,l=1}^{N} A_{kl}B_{kl}. \quad (14)$$

Based on this, the distance correlation between X and Y is defined as follows.

$$d(X, Y) = \begin{cases} \dfrac{\nu(X, Y)}{\sqrt{\nu(X, X)\nu(Y, Y)}} & \text{if } \nu(X, X)\nu(Y, Y) \neq 0, \\ 0 & \text{else.} \end{cases} \quad (15)$$

Then, the distance correlation between X and Y is measured as $d(X, Y)$.

Some of the mathematical properties of distance correlation are

(1) $\nu(X, Y) \geq 0$;

(2) $0 \leq d(X, Y) \leq 1$;

(3) $d(X, Y) = 0$ if and only if X and Y are independent

(4) $d(X, Y) = 1$ implies that X and Y have the equal dimensionality and $Y = A + bCX$, wherein A is a vector, b is a real number, and C is an orthonormal matrix

2.1.2. Advantages of Distance Correlation. First, from the definition of distance correlation, we can get that its most important advantage is that it can measure correlation between multidimensional vectors.

Additionally, distance correlation can illustrate correlation between vectors more accurately than most of existing methods. We use an example shown in Figure 2 to compare the correlation between X and Y got by distance correlation method and the other two important methods—PCC and COS, respectively. Figure 2 presents two vectors, $X = \{x_n : n = 1, 2, \cdots, N\}$ wherein $x_i = N/2 - i$ if $i \leq N/2$ else $x_i = i - N/2$; and $Y = \{y_n : n = 1, 2, \cdots, N\}$ wherein $y_i = i - N$. We get from Figure 2 that the value of X allows for a nice estimation of the value of Y, vice versa. In another words, X and Y are very relevant to each other. However, the correlation between X and Y is 0 measured by PCC and 0.75 by COS. As the comparison, the correlation between X and Y is 1 measured by distance correlation.

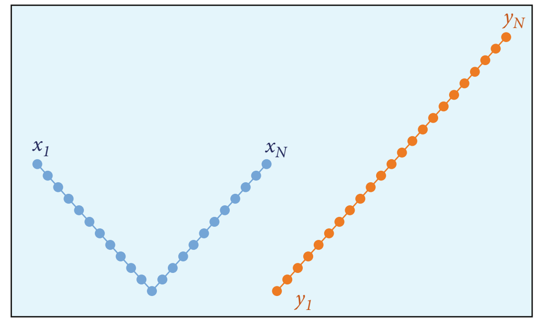

Figure 2: Two vectors: $X = \{x_n : n = 1, 2, \cdots, N\}$, $Y = \{y_n : n = 1, 2, \cdots, N\}$.

2.2. Enhance Distance Correlation by Time Factor. It is reasonable to consider that time can affect the correlation between two systems. It often means that the later the data is obtained, the more significant the impact on the correlation, vice versa. Therefore, this section combines time factor and distance correlation to measure correlations more accurately. The improved distance correlation is identified as MOCC correlation.

If two systems were observed in the same time series of length N, and let the 1st observation is the earliest, then, the N-th observation should have the largest weight on their correlation, and the $(N-1)$-th has the second largest. Therefore, if we bring time factor into distance correlation, equation (14) is revised as

$$\nu^2(X, Y) = \frac{1}{N^2}\sum_{k,l=1}^{N} (k + l)A_{kl}B_{kl}, \quad (16)$$

wherein $(k + l)$ is the time factor. The bigger k or l indicates the later time, therefore, has the bigger weight. Introduce $\nu^2(X, Y)$ of equation (16) into equation (15), the MOCC correlation will be got.

3. Applications of MOCC Correlation

A significant application of MOCC correlation is missing data prediction, which is also widely studied in the research field of IoT systems. MOCC correlation is a transform of the distance correlation. Although distance correlation is widely adopted by many researchers, the missing value prediction based on distance correlation has rarely been studied before.

An IoT system often consists of massive sensors, and data missing is a common phenomenon for system running. Data missing may be caused by network packet loss, or some sensors' transient exception. If it occurs, then some data in Table 1 will be missing, the prediction of missing data for an IoT system consists of the following two steps: (1) finding its high correlated systems based on history data and (2) predicting missing value based on its high correlated systems.

3.1. High Correlated System Finding. For ease of presentation, denote the set of all the IoT systems as $S = \{S_1, S_1, \cdots, S_N\}$, if one system has data missing, its high correlated systems have to be found for missing data prediction. Denote

S_a is the system that needs missing data prediction, the set of its all high correlated systems can be denoted as:

$$S' = \{S_k : d(S_a, S_k) \geq \text{THRESHOLD}, S_k \in S\}, \quad (17)$$

wherein $d(S_a, S_k)$ is the MOCC correlation between system S_a and S_k, THRESHOLD is a constant to indicate whether these two systems are high correlated or not. In practice, we let THRESHOLD = 0.8.

3.2. Missing Data Prediction. Before the missing data prediction, we should determine another essential issue: how a system is impacted by its high correlated systems. By experiments, we found two phenomena:

(1) If two systems have a high MOCC correlation, then, their MOCC correlation will hardly change with their data amount growing with time

(2) On the contrary, if two systems have only a low MOCC correlation, then, their MOCC correlation would change notably with their data amount growing

This experiment will be detailed and presented in Section 4, and this section only use the corollary of this experiment to predict missing data. From the above phenomena, we can get a corollary as follows.

Corollary 1. *If S_a is an IoT system with missing data and S_k is high correlated to S_a, then, the prediction of the missing data should keep $d(S_a, S_k)$ almost unchanged.*

Take the IoT system in Table 1 as an example, v_{mn} is the observed value of Sensor$_m$ on Time$_n$, if this value is missing, denote the prediction of it as $\hat{v}_{mn}$.

Based on the observations of S_a and S_k from Time$_1$ to Time$_{n-1}$, we have got their MOCC correlation denoted as $d^{(n-1)}(S_a, S_k)$. If the observation on Time$_n$ missed the value v_{mn}, then, the prediction $\hat{v}_{mn}$ should satisfy

$$d^{(n)}(S_a, S_k) \approx d^{(n-1)}(S_a, S_k), \quad (18)$$

wherein $d^{(n)}(S_a, S_k)$ is got by predicting the missing v_{mn} as $\hat{v}_{mn}$.

If we denote S_a as X, and its high correlate system S_k as Y, after the observations on Time$_{n-1}$, we rewrite equation (14) as

$$\nu^{(n-1)}(X, Y) = \sqrt{\frac{1}{(N-1)^2}\sum_{k,l=1}^{N-1} A_{kl}B_{kl}}. \quad (19)$$

By predicting the missing v_{mn} as $\hat{v}_{mn}$, namely, that we have completed the observations on Time$_n$, if the following satisfied

$$\nu^{(n-1)}(X, Y) \approx \nu^{(n)}(X, Y) = \sqrt{\frac{1}{N^2}\sum_{k,l=1}^{N} A_{kl}B_{kl}}, \quad (20)$$

then, equation (20) will be satisfied. Finally, we can get that the value that satisfied equation (22) is the needing prediction of v_{mn}. Although equation (22) seems complicated, it is just a multiple-order equation with only one variable $\hat{v}_{mn}$, we can solve it quickly with the facility of three-part tool, such as Apache Commons math library.

For the missing value of v_{mn}, we can get different predictions based on its different high correlated system. We can make the prediction more reasonably by combining all the results of all its correlated system in S_h, shown as follows:

$$\hat{v}_{mn} = \frac{1}{\sum_{S_k \in S'} d(S_a, S_k)} \sum_{S_k \in S'} d(S_a, S_k) \bullet \hat{v}^{(k)}{}_{mn}, \quad (21)$$

wherein, $\hat{v}^{(k)}{}_{mn}$ is the prediction value made by system S_k, and $\hat{v}_{mn}$ is the final prediction value of v_{mn}.

4. Experiments

In this section, we perform experiments to validate our MOCC method and compare the results with those from other correlation computing methods. Our experiments are intended to (1) verify the rationality of Corollary 1 that is presented in Section 3.2 and (2) compare efficiency of MOCC method with other correlation computing methods.

4.1. Experiment Setup. This experiment was constructed by employing the data of a typical IoT system—the FY-3D weather satellite, one of the most advanced weather satellites globally. This satellite consists of energy, attitude, propulsion, thermal control, data transmission, cabin, and payload seven subsystems. Each subsystem can also be divided into smaller systems; for example, the attitude control consists of stellar positioning, gyroscope, and flywheel systems. There are more than 10000 sensors deployed on this satellite. All the sensors' monitoring data are transferred to the ground station periodically fourteen times one day.

It is important to work to analyze the correlation between two subsystems of the satellite since one's status often impacts another one. Satellite communications are vulnerable to interference. Therefore, data missing or abnormality usually occurs. This motivated the work of this paper.

4.2. Experimental Proof of Corollary 1. In Section 3.2, Corollary 1 says: if S_a is an IoT system with missing data and S_k is high correlated to S_a, then, the prediction of the missing data should keep $d(S_a, S_k)$ almost unchanged.

To prove this corollary, we have made statistics of all subsystems of FY-3D weather satellite for their correlations. For any two subsystems, we called it a system pair. This experiment is constructed as the following six steps:

(1) Determine a period with no missing data; use this data to do steps 2 to 6

(2) Calculate the MOCC correlation of any two systems, and determine their correlation belongs to which range. There are 10 ranges in total, i.e., [0, 0.1)… [0.9, 1]

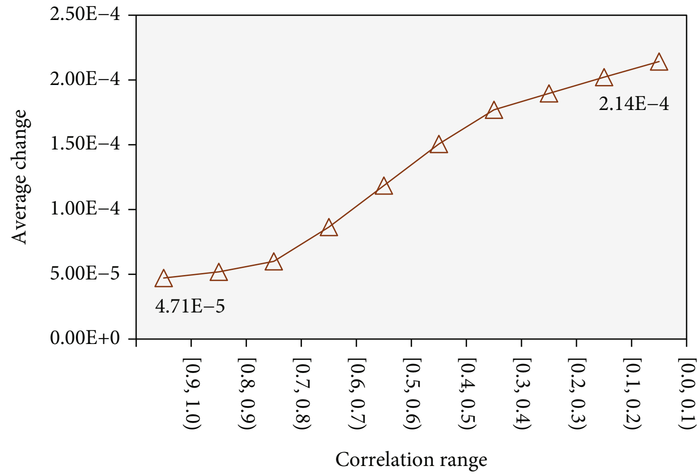

Figure 3: Average change in each correlation range, which proves Corollary 1.

Table 2: Accuracy comparison.

	Methods	Data = 5%		Data = 10%		Data = 20%		Data = 50%	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Voltage	TA	0.612	1.564	0.595	1.533	0.593	1.509	0.483	1.471
	MF	0.563	1.465	0.494	1.271	0.445	1.174	0.397	1.071
	MOOC	0.379	0.938	0.305	0.753	0.225	0.557	0.122	0.312
Current	TA	27.286	75.535	17.064	51.697	14.948	50.519	14.837	47.409
	MF	20.209	54.794	16.158	46.742	14.006	41.695	14.752	41.467
	MOOC	17.831	51.712	13.125	42.333	10.026	36.209	9.904	35.084
Temperature	LA	20.085	59.816	15.175	45.563	11.901	37.139	10.459	29.518
	MF	16.488	50.151	12.894	36.755	10.919	34.798	9.854	25.112
	MOOC	8.289	24.752	6.297	16.197	5.001	14.096	3.576	11.144

(3) For each system pair, randomly remove some data observed simultaneously. It can be seen to delete a column from Table 1 randomly, but the deletion of two tables should be at the same column position. The deletions were finished only 5% columns left. (Since we cannot know the actual values of the future, we deleted some values from the original dataset. Then, the dataset that deleted more values can be seen as “the past,” and the dataset that deleted fewer values can be seen as “the future.” By which, we can simulate the change of two systems’ correlation along with time.)

(4) Once a deletion for a system pair finished, we calculated their new MOCC correlation and got the change value comparing their last correlation

(5) When all the deletions were finished, the average change of each system pair was got

(6) We find that if a system pair originally belonged to a relatively large range in step 2, then, their correlation would rarely change when deleting their data. The experimental result is shown as Figure 3

4.3. Comparisons. We compare the predictive accuracy for missing data of our MOCC method with other two well-known prediction methods. The two compared methods are the follows:

(1) *Time-Aware Method (TA)*. This type of methods makes prediction based on the time factor, which was proposed in reference [16]

(2) *Matrix Factorization Based Method (MF)*. This type of methods makes prediction by factorizing the dataset into matrices, and the reconstruct the dataset by multiplying these matrices. We selected the method proposed in reference [17] to compare with

We use the mean absolute error (MAE) and root mean squared error (RMSE) to measure the prediction accuracy. MAE and RMSE are defined as (22) and (23), respectively:

$$\text{MAE} = \frac{\sum |v_{mn} - \hat{v}_{mn}|}{N}, \tag{22}$$

$$\text{RMSE} = \sqrt{\frac{\sum (v_{mn} - \hat{v}_{mn})^2}{N}}, \tag{23}$$

where v_{mn} is a value in the dataset, and $\hat{v}_{mn}$ is its prediction value.

The predictions were made as follows:

(1) Randomly select some values which are not missing to predict, such we can compare the predictive value to the real value

(2) The dataset was made to different sparse ratio to test the prediction performance on spare data

There are many subsystems deployed on FY-3D satellite, we choose the battery system to make comparisons, and this system consists of the following sensors: (1) voltage, (2) current, and (3) temperature.

The prediction accuracies of MOCC and the comparisons with other methods are shown in Table 2. With reference to Table 2, we can see that MOCC is more accurate than all of the other methods for the two chosen datasets. As the data increases from 5% to 50%, the MAE and RMSE values become smaller.

5. Conclusion

We have enhanced the concept of distance correlation by bringing the time factor into it, which results in our MOCC method. This method considers all sensors of an IoT system as integration and can measure the correlation between subsystems accurately. We have also presented how to predict missing data values by our MOCC method. The prediction is based on an experimental proofed corollary, i.e., two highly correlated systems will rarely change their correlation with the data amount growing.

We also performed experiments to verify our corollary and our method's efficiency.

6. Related Work

Correlation between systems indicates their dependency, which is very important in system analysis. Based on correlation computing, we can determine the most relevant factor of the systems subsystem. For example, we can find that a system's data anomaly will affect other systems' functional status if there are high correlations between them; or a system's data missing can be predicted based on its high relevant systems. IoT systems consist of massive sensors producing data in matrix form. Therefore, the correlation must be calculated in multiple-dimensional ways.

There are many existing correlation computing methods, such as Pearson correlation coefficient (PCC) [8] and cosine coefficient (COS) [9], and new correlation measuring methods have been presented in recent years [4]. Many enhancements were brought into PCC and COS in recent years. By adding the weights to determine the different effects of different correlated objects, Zheng et al. [18] proposed an improved PCC correlation computation method and employed this method to predict missing data. Sun et al. [19] proposed a normalized correlation computing method to avoid the disadvantage that traditional PCC or COS neglects the mathematical features of observed vectors. However, all existing methods can only measure the correlation between one-dimensional vectors but not matrices. Figure 1 compares the vector and matrix-oriented similarity measurements.

Prediction of missing data has been widely studied in many fields, especially in the field of QoS prediction for service recommendation [20–22]. Correlation analysis is a crucial way to make a prediction. However, the missing value prediction based on distance correlation has rarely been studied before this paper.

AI methods such as neural networks were introduced to resolve multidimensional correlation analysis in recent years [13]. However, these black-box approaches cannot explain the mathematical mechanisms, and lots of memory or time are consumed.

Data Availability

Data are available on request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key Technology Research and Development Program of China (2021YFB3901000 and 2021YFB3901005) and NSFC (61602126).

References

[1] J. Singh, T. Pasquier, J. Bacon, H. Ko, and D. Eyers, "Twenty security considerations for cloud-supported internet of things," *IEEE Internet of Things Journal*, vol. 3, no. 3, pp. 269–284, 2016.

[2] F. Gottwalt, E. Chang, and T. Dillon, "CorrCorr, a feature selection method for multivariate correlation network anomaly detection techniques," *Computers & Security*, vol. 83, pp. 234–245, 2019.

[3] A. Rawat, A. Gupta, A. Singh, and S. Bhushan, "Energy conservation and missing value prediction model in wireless sensor network," in *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, pp. 1–5, Ghaziabad, India, April 2019.

[4] B. Liu, C. Gong, and Z. Xu, "Correlation analysis and link gain prediction for optical wireless scattering communication over broad spectra," *IEEE Transactions on Wireless Communications*, vol. 19, no. 2, pp. 1386–1396, 2020.

[5] K. Zhang, F. Zhao, S. Luo, Y. Xin, and H. Zhu, "An intrusion action-based IDS alert correlation analysis and prediction framework," *IEEE Access*, vol. 7, pp. 150540–150551, 2019.

[6] X. Feng, X. Ling, H. Zheng, Z. Chen, and Y. Xu, "Adaptive multi-kernel SVM with spatial–temporal correlation for short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2001–2013, 2019.

[7] H. Yang, L. Gao, and G. Li, "Underwater acoustic signal prediction based on correlation variational mode decomposition and error compensation," *IEEE Access*, vol. 8, pp. 103941–103955, 2020.

[8] U. Shardanand and P. Maes, “Social information filtering: algorithms for automating word of mouth,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 210–217, Denver USA, May 1995.

[9] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in *Proceedings of the 10th international conference on World Wide Web*, pp. 285–295, Hong Kong, April 2001.

[10] S. Beigi and A. Gohari, “Φ -entropic measures of correlation,” *IEEE Transactions on Information Theory*, vol. 64, no. 4, pp. 2193–2211, 2018.

[11] H. Aly and A. Winterhof, “A note on Hall’s sextic residue sequence: correlation measure of order k and related measures of pseudorandomness,” *IEEE Transactions on Information Theory*, vol. 66, no. 3, pp. 1944–1947, 2020.

[12] J. Levin and G. Smith, “Optimized measures of bipartite quantum correlation,” *IEEE Transactions on Information Theory*, vol. 66, no. 6, pp. 3520–3526, 2020.

[13] H. Chen, Z. Chen, Z. Chai, B. Jiang, and B. Huang, “A single-side neural network-aided canonical correlation analysis with applications to fault diagnosis,” *IEEE Transactions on Cybernetics*, pp. 1–13, 2021, https://ieeexplore.ieee.org/document/9376288.

[14] G. J. Szekely, M. L. Rizzo, and N. K. Bakirov, “Measuring and testing dependence by correlation of distances,” *The Annals Of Statistics*, vol. 35, no. 6, pp. 2769–2794, 2007.

[15] G. J. Szekely and M. L. Rizzo, “Partial distance correlation with methods for dissimilarities,” *The Annals of Statistics*, vol. 42, no. 6, pp. 2382–2412, 2014.

[16] Y. Jin, W. Guo, and Y. Zhang, “A time-aware dynamic service quality prediction approach for services,” *Tsinghua Science and Technology*, vol. 25, no. 2, pp. 227–238, 2020.

[17] J. Zhu, W. Ma, and Y. Song, “Attentive matrix factorization for recommender system,” in *2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pp. 932–936, Chengdu, China, October 2020.

[18] Z. Zheng, H. Ma, M. R. Lyu, and I. King, “QoS-aware web service recommendation by collaborative filtering,” *IEEE Transactions on Services Computing*, vol. 4, no. 2, pp. 140–152, 2011.

[19] H. Sun, Z. Zheng, J. Chen, and M. R. Lyu, “Personalized web service recommendation via normal recovery collaborative filtering,” *IEEE Transactions on Services Computing*, vol. 6, no. 4, pp. 573–579, 2013.

[20] W. Ma, Q. Zhang, M. Chunxiao, and M. Zhang, “QoS prediction for neighbor selection via deep transfer collaborative filtering in video streaming P2P networks,” *International Journal of Digital Multimedia Broadcasting*, vol. 2019, Article ID 1326831, 10 pages, 2019.

[21] Y. Ma, S. Wang, P. C. K. Hung, C. Hsu, Q. Sun, and F. Yang, “A highly accurate prediction algorithm for unknown web service QoS values,” *IEEE Transactions on Services Computing*, vol. 9, no. 4, pp. 511–523, 2016.

[22] S. Wang, Y. Ma, B. Cheng, F. Yang, and R. N. Chang, “Multi-dimensional QoS prediction for service recommendations,” *IEEE Transactions on Services Computing*, vol. 12, no. 1, pp. 47–57, 2019.