Computers, Materials & Continua CMC, vol.64, no.2, pp.987-1004, 2020

Three-Phase Unbalance Prediction of Electric Power Based on Hierarchical Temporal Memory

Hui Li[1], Cailin Shi[2, 3], Xin Liu[2, 3], Aziguli Wulamu[2, 3, *] and Alan Yang[4]

Abstract: The difference in electricity and power usage time leads to an unbalanced current among the three phases in the power grid. The three-phase unbalanced is closely related to power planning and load distribution. When the unbalance occurs, the safe operation of the electrical equipment will be seriously jeopardized. This paper proposes a Hierarchical Temporal Memory (HTM)-based three-phase unbalance prediction model consisted by the encoder for binary coding, the spatial pooler for frequency pattern learning, the temporal pooler for pattern sequence learning, and the sparse distributed representations classifier for unbalance prediction. Following the feasibility of spatial-temporal streaming data analysis, we adopted this brain-liked neural network to a real-time prediction for power load. We applied the model in five cities (Tangshan, Langfang, Qinhuangdao, Chengde, Zhangjiakou) of north China. We experimented with the proposed model and Long Short-term Memory (LSTM) model and analyzed the predict results and real currents. The results show that the predictions conform to the reality; compared to LSTM, the HTM-based prediction model shows enhanced accuracy and stability. The prediction model could serve for the overload warning and the load planning to provide high-quality power grid operation.

Keywords: Three-phase unbalance, power load, prediction model, hierarchical temporal memory.

1 Introduction

With the development of power information technology, the construction of a smart grid is advancing. Data collected is increasing in size based on marketing business applications, electricity information collection, power management systems, etc. [Zhu and Yang (2019)]. Three-phase unbalance refers to the inconsistency of the three-phase current

[1] School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing, 100083, China.

[2] School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, 100083, China.

[3] Beijing Key Laboratory of Knowledge Engineering for Materials Science, Beijing, 100083, China.

[4] Amphenol Assemble Tech, Houston, TX 77070, USA.

[*] Corresponding Author: Aziguli Wulamu. Email: aziguli@ustb.edu.cn.

Received: 20 January 2020; Accepted: 14 April 2020.

amplitude in the power system, and the amplitude difference exceeds the specified range. The occurrence of three-phase unbalance is related not only to user load characteristics but also to power system planning and load distribution. Therefore, it is essential for safe and high-quality power supply to deeply analyze the three-phase unbalance problem in the transformer section of 10 kV transmission line.

To solve this problem, we choose the Hierarchical Temporal Memory model to analyze the three-phase unbalance problem, mainly to predict the current and to achieve the effect of three-phase unbalanced monitoring and early warning.

Hierarchical Temporal Memory (HTM) is a technology modeled on how the neocortex performs these functions. HTM offers the promise of building machines that approach or exceed human-level performance for many cognitive tasks [Numenta (2011)]. HTM is a continuous learning system derived from theory the of neocortex [Hawkins, Ahmad and Dubinsky (2010)] which is a biologically inspired machine intelligence technology that mimics the architecture and process of the neocortex [Poirazi, Brannon and Mel (2003); Polsky, Mel and Schiller (2004)]. HTM is well suited for real-time applications.

Time series data refers to the data with time sequence relation, which frequently appears in the time series database [Yu, Liu, Wang et al. (2018)]. The current value is typical time-series data.

Before deep learning, many forecasting methods were developed to address time series data such as Auto-Regressive Model, Moving Average Model, Auto-Regressive Integrated Moving Average Model, and others. The long short-term memory (LSTM) [Hochreiter and Schmidhuber (1997); Mohamed and Karar (2019); Fazle, Somshubra, Houshang et al. (2019); Assaf and Tammy (2019); Liu, He, Li et al. (2019); Kouba, Menaa, Tehrani et al. (2019)] algorithm has achieved excellent results in the processing of time series data. In Zhou's study [Zhou (2018)], which is the relevant result of the laboratory, multiple variables such as temperature, relative humidity, etc., were used to the LSTM algorithm of current prediction.

However, the network structure and parameter adjustment of deep learning are unexplained. With the arrival of the brain research and cognitive neuroscience era, HTM begins to emerge as a novel biological neural network. HTM has been used to address various learning tasks in different domains. For example, they have been used to predict visual target sequences [Kirtay, Falotico, Ambrosano et al. (2016)], to establishing individual memory-predictive models for rumor communication [Wang (2013)], to process bio-signals and predict sensory and location data in the context of smart homes [Otahal and Stepankova (2014)], to model typical geospatial travel patterns and identify anomalies in movement [Numenta (2011)], to detect anomalies in publicly traded stocks [Numenta (2011)], to recognize the optic nerve in retina images [Boone, Karnowski, Chaum et al. (2010)] and to build a commercial proactive and automatic incident response system called Grok, among other applications.

HTM has the following advantages:

1) HTM is naturally very noise tolerant, so the whole system could be a fault-tolerant.

2) HTM is unsupervised, so it does not need labeled data.

3) Temporality is a crucial component of recognizing real-world patterns. Especially

predicting what will happen next. HTM is about solving temporal patterns. HTM is naturally time-variant since it is stately and context-dependent.

The main contributions of this paper are summarized as follows:

1) We applied the Hierarchical Temporal Memory model on the current prediction to achieve the effect of three-phase unbalanced early warning and monitoring. It helps business to formulate targeted solutions, ensure stable operation of the power grid, reduce transformer losses, improve transformer safety index, and provide high-quality power services.

2) Through the research in this paper, the three-phase unbalance (current) of the heavy overload zone under the 10 kV line will be predicted more accurately.

3) Comparing the prediction accuracy of HTM with LSTM. The result is that HTM predictions are more accurate and stable in general.

2 Sparse distributed representations

A Sparse Distributed Representations (SDR) consists of a vast array of bits of which most are zero, and a few are one. Each bit carries some semantic meaning, so if two SDRs have more than a few overlapping on-bits, then those two SDRs have similar meanings [Purdy (2016)].

Neuroscientists found that neocortex represents information using sparsely distributed patterns of activity [Barth and Poulet (2012)]. Refer to SDR as the data structure of the brain. For example, when playing a musical instrument, some of the neurons in the auditory cortex are becoming active when they hear specific frequencies, but most of them are silent. The same goes for the visual system. In every sensor region, there is a sparse pattern of activity that represents the perception of the world at any time point.

Essentially SDRs are used in the cortex for every aspect of cognitive function for every sensory modality:

$$\binom{n}{w} = \frac{n!}{w!(n-w)!} \tag{1}$$

Because of the nature of the factorials in Eq. (1) the capacity of SDRs is going up very quickly. A sparsity of 2%, the SDRs will have enormous capacity. Inside of an SDR, it can be fitted with a massive amount of data and represent an enormous amount of different values.

In our algorithm, we added some noise to the SDRs to show its tolerance. We set n=2048, w=41, and theta=30; theta is the threshold to determine if the original SDRs match the SDRs with noise. That means if there are over 30 bits the same with the original, the original SDRs match the SDRs with noise. In this situation, the SDRs can tolerate almost up to 29% noise, and the chance of false-positive is 7.772×10^{-51}. This proves that there will be a small minuscule percentage that this will be a false match.

A neocortical pyramidal neuron has thousands of excitatory synapses located on dendrites (inset). There are three sources of input to the cell: the feed-forward inputs (shown in green), the more distal basal and apical dendrites.

A pyramidal neuron in the neocortex is a complex pattern recognition system, and it is receiving a constant stream of SDRs. As Fig. 1 shows, it is getting SDRs from its apical dendrites, which are receiving feedback from higher levels in the hierarchy those connections to cells up at a higher level. It is receiving SDRs from basal dendrites that are like contextual SDRs coming from cells in the same region of the hierarchy. And it is also receiving SDR input from proximal dendrites, and these are coming from the lower levels of the hierarchy, their direct connections to cells sensory input so every one of these pyramidal cells are constantly receiving a stream of lots of different SDRs through its dendrites, and all it is doing is trying to decide when it is going to fire. When it fires, it plays a role in all the thousands of SDRs where it is represented by turning its bits on.

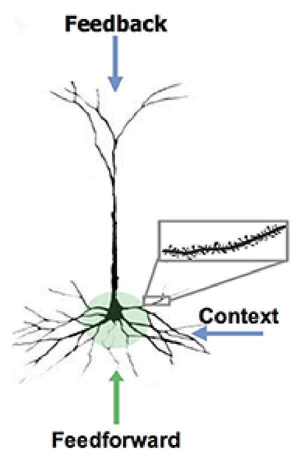


Figure 1: The dendritic tree of pyramidal neurons [Cui, Ahmad and Hawkins (2017)]

The data used in this paper includes the current value of IA, IB, IC three-phase daily 96 points, and data from January 2017 to September 2017 in some of the sub-commutations in the five cities of North Hebei region. This data is derived from the electricity information collection and marketing business application system. The data volume information is as Tab. 1 shows.

Table 1: Sample data in different cities

Cities	Total number of records
Tangshan	20249
Langfang	24944
Qinhuangdao	24944
Chengde	24944
Zhangjiakou	25080
Total	120161

These data will be transformed into SDRs with these advantages after being passed through encoders and spatial pooler described in detail below.

3 Hierarchical temporal memory

Hierarchical Temporal Memory (HTM) is a machine learning technology that aims to capture the structural and algorithmic properties of the neocortex [Numenta (2011)]. In many machine learning algorithms, HTM is outstanding because of the following reasons:

1) HTM system uses an HTM Neuron. The HTM Neuron treats distal input differently than proximal input. It implements the effects of a dendritic spike because it allows some localized apical/distal input to affect how the cell fires in response to proximal input. The detailed description is referred by Hawkins et al. [Hawkins and Ahmad (2016)].

2) HTM system requires high-dimensional spaces to represent synapses. In most of the HTM implementations, binary Sparse Distributed Representations (SDRs) are needed to fill this need. Sensory input must be encoded into high-dimensional space with semantic meaning.

3) The model contains layers of HTM Neurons used as compute modules, which can use different data sources for proximal, distal, and apical input. Each layer's output can be input for another layer.

The main flow chart of HTM prediction model is shown in Fig. 2. An HTM prediction model has four primary elements: Encoders, Spatial Pooler, Temporal Memory, and SDR Classifier.

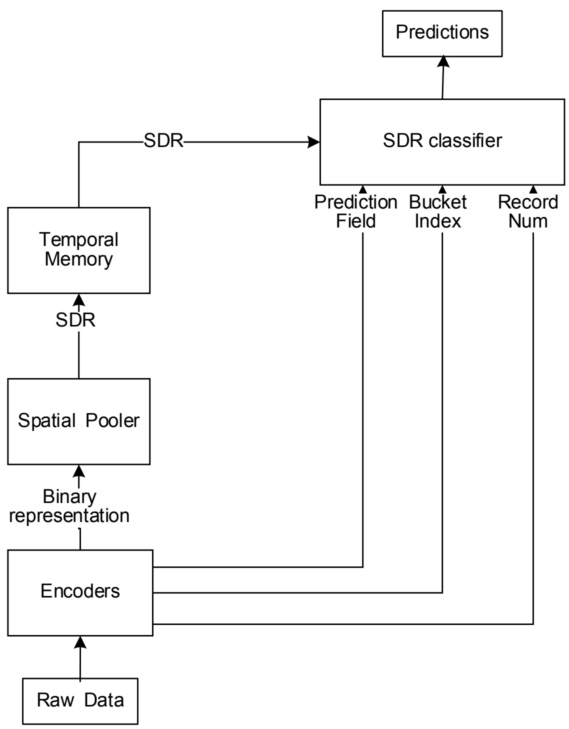


Figure 2: The main flow chart of HTM prediction model

3.1 Encoders

The application of HTM is quite extensive if the data can be converted into an SDR. Therefore, using an encoder to convert raw data into binary representations is always the first step of applying the HTM model. The encoder converts the native format of the data into an SDR that can be fed into an HTM system [Purdy (2016)]. The encoder will make input data easy to capture important semantic characteristics.

An encoder is like the outermost layer of an HTM system that translates real-world data into SDR so that it can be processed by HTM systems. Given input value must be converted in a way that captures the essential semantic characteristics of data inside the SDR that is being created.

According to Purdy's article [Purdy (2016)], there are four Principles of Encoding data:

1) Semantically similar data should result in SDRs with overlapping active bits.

2) The same input should always produce the same SDR as output.

3) The output should have the same dimensionality (total number of bits) for all inputs.

4) The output should have similar sparsity for all inputs and have enough one-bits to handle noise and subsampling.

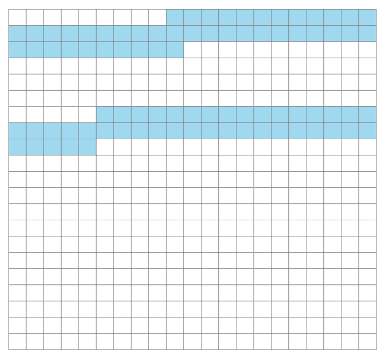


Figure 3: Example of current scalar encoding

As shown in Fig. 3, according to the data that will be used in this article, one set of current data has a maximum value of about 400 and a minimum value of about 0. We can calculate:

$$range = maximum - minimum \tag{2}$$

The value obtained is 400. So we set n=441, w=42, The relationship between n and w is as follows:

$$n = range + w - 1 \tag{3}$$

The blue cell represents the bit is on while the white cell represents the bit is off. From 10th to 51th is one bucket, and it is basically for the scalar encoder. It is a consecutive array of on bits of exactly 42-length because we set the w=42 at the beginning. Given that we want a bucket width of 42, it can fit 441 of those in this space.

000000000111. . . 111000. . . 0

10th

0000000000111. . . 111000. . . 0

11th

Figure 4: The current value represented in binary form

When the 10th to 51st cell is on, it represents the current value 10, when the 11th to 52th is on, it represents the current value 11. It is represented in binary form, as shown in Fig. 4. The difference between the value is basically that these couple of bits moved from one side to the other, but they are semantically similar. It is evident that they have a significant amount of overlap that is what makes 10 identical to 11. For another instance, there is no overlap at all between the two representations.

This is the most common method of doing number encoding for the current HTM systems.

3.2 Spatial pooler

It is known that a neocortical neuron has thousands of synapses located on dendrites [Cui, Ahmad and Hawkins (2017)]. A few synapses close to the cell body, called proximal synapses, have a relatively large effect on a cell. Most synapses are far from the cell body, and they are called distal synapses [Wu, Zeng, Chen et al. (2017)]. The proximal zone receives the feed-forward input. The distal basal zone receives contextual information from nearby cells in the same cortical region [Spruston (2008)].

In an end-to-end HTM system, the spatial pooler transforms input patterns into SDRs in a continuous online fashion [Cui, Ahmad and Hawkins (2017)].

There are two most significant goals of spatial pooling, one is for the spatial pooler to maintain a fixed sparsity, so as it sees input in the input space no matter how many bits are on in it, the spatial pooler needs to maintain a certain sparsity when it outputs all the time. The second goal probably is the most important one that the spatial pooler must maintain the overlap properties of the input data, so in this input space, if there are two different representations over time and they have a high overlap score, it means that they are semantically similar the output data that the spatial pooler creates. To represent those pieces of data also must have a high overlap score, so if there are two similar inputs Then we should get two related outputs.

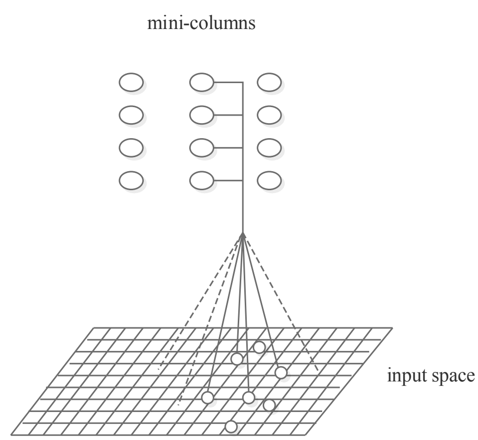


Figure 5: Proximal dendric connections. The blue dots are cells, and every four cells form a set of mini-columns. Each column is connected to different bits in the lower grid, which represents input space

As Fig. 5 shows, there are three mini-columns, and each of them has four cells. Each column has a different potential pool of input cells that it might be connected to. Each one of those potential connections also has a permanence attached to it. The mini-column has a receptive field, so each column is connected to different bits in the input space. These are proximal dendric connections feed-forward input into the system, and every cell shares this proximal input in each column. Every column has a separate receptive field, and each one of the cells within the column shares that receptive field through its proximal connection. The spatial pooler takes that input space and translates the incoming data into active columns in its representation in the output space.

A local inhibition mechanism implements among neighboring mini-columns. It ensures that a small fraction of the spatial pooler (SP) mini-columns that receive most of the inputs are active within the local inhibition radius. Synaptic permanency is adjusted according to the Hebbian rule [Hebb (1988)]: for each active SP mini-column, active inputs (black lines) are reinforced, and inactive inputs (dashed lines) are punished [Cui, Ahmad and Hawkins (2017)].

3.3 Temporal memory

The HTM temporal memory learns temporal sequences of these SDRs and makes predictions for future inputs [Cui, Ahmad and Hawkins (2016)].

HTM sequence theory. The design of the mini-columns model usually contains multiple cells in the HTM. The prediction process and results of one cell and multiple cells are different. Suppose we use the current data with the following sequence, the detailed process can be seen in this example.

			102				102				102										102	102				102				
		98				98				98													98		98					
	96				96				96			96		96		96	96			96				96			96			
															95			95											95	
94				94				94					94						94									94		94

Figure 6: An example of the current sequence. The number appearing in each grid is the current value. This graph shows the current value at each moment as time increases uniformly

Single order memory just goes through the whole current sequence in Fig. 6. Look for all the value 96 and keep count of what follows done how many times that way. Based on the calculation result, 98 is going to follow the value 96 in a sequence.

The high order memory uses the currently plus what is happened previously; it can remember more about the notes that led up to the current state. So, if the value 102 is followed by 96, it can more accurately predict what comes next, and in this case, 102 and 96 will be followed by 94.

So, having one cell per column restricts each column to respect a spatial feature in only one context meaning, it only has information from the current state to make predictions about the feature. This severely limits the functionality as a learning algorithm. A good sequence memory needs to recognize spatial patterns with many different contexts, and that is why the HTM model needs many cells for mini-column.

Temporal memory algorithm. The temporal memory algorithm is mainly to complete two tasks [Hawkins (2016)]. One, it learns the sequences of active columns in the spatial pooler over time. And two, it makes predictions about what pattern is coming next based on the temporal context of each input. It does this by activating individual cells within many columns.

There are two primary phases of the temporal memory algorithm, and the first is to identify which cells within active columns will become active on the time step, the second phase once those activations have been identified is to choose a set of cells to put into a state this means that these cells will be primed to fire on the next time step.

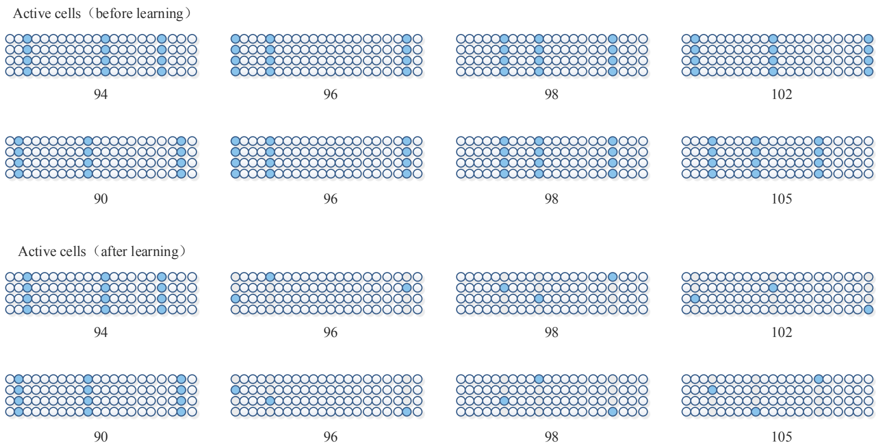


Figure 7: An example of current sequence memory. The two rows of figures in the upper half and the lower half represent the activation of cells before and after learning, respectively. Solid dots indicate that the cell is activated. Numbers from left to right represent input data

Fig. 7. is an example of current sequence memory, consider two current sequences in the input data 94-96-98-102 and 90-96-98-105. All the columns burst because there is no context for 94. It is always the first value in the sequence. Then the next pattern for 96 comes, and we identified not only that this is a spatial pattern for ‘96’, but it is within the context of the spatial pattern ‘94’ so we are going to call this 96’, it is not just 96, it is 96 with an additional temporal component. The same thing with 98’, which is the spatial representation for 98 with the additional context of 98 in the context of 96’ in the context of 94, so we are driving sort of down this temporal sequence, and it goes all the way to 102 which is now 102’.

3.4 SDR classifier

The purpose of the classifier is learning associations between a given state of the temporal memory at time t, and the value that is to be fed into the Encoder at time t+n. What SDR Classifier does is mapping activation patterns to probability distributions, which is mapping vector of temporal memory’s active cells to the possible encoder buckets.

HTM is fundamentally a memory-based system. HTM networks are trained on lots of time-varying data and rely on storing a large set of patterns and sequences.

The steps of prediction algorithm using a hierarchical real-time memory model are shown in Tab. 2:

Table 2: Prediction algorithm of HTM

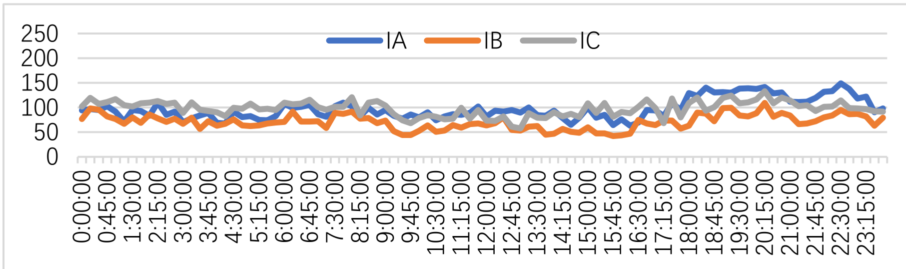
Algorithm 1 HTM prediction algorithm
Input:
Training sample set X
Output:
Prediction results
Step 1: Encoding
1) Select encoding interval, minimum value, and maximum value
2) The calculation encoding range=minimum value–maximum value
3) Select the number of buckets for the subpackage data as buckets
4) Select the number of active bits in each characterization as w
5) Calculate the total number of digits as n=buckets+W–1
6) For the given value v, select the dropped sub-container: i=floor [buckets×(v–minimum value)/range]
Step 2: Spatial Pooler
1) Initialization of spatial sedimentation tank: calculate the overlap value between each microcell column and the current input, and multiply the overlap value with the promotion coefficient
2) Calculate the winning micro cell column set after inhibition
3) Update synaptic connectivity and other internal variables
Step 3: Temporal Memory
1) Calculate the active state of each cell. When a cell is active, it will establish a connection with the cells that are active before it;
2) Calculate the predicted state of each cell. Cells can predict when they will be activated by focusing on their connections;
3) Update the state of the neural connection.

4 Experiment

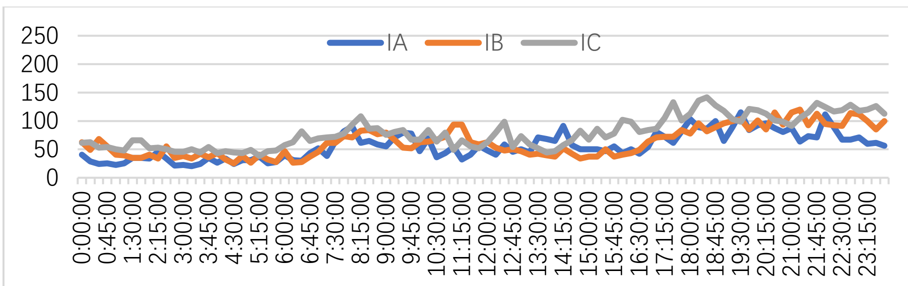
System data is collected based on marketing business applications, electricity information collection, PMS, etc. We use a hierarchical memory model to predict the current of the public substation in 10 kV transmission line. It is mainly for early warning monitoring of three-phase unbalance.

4.1 Raw data analysis

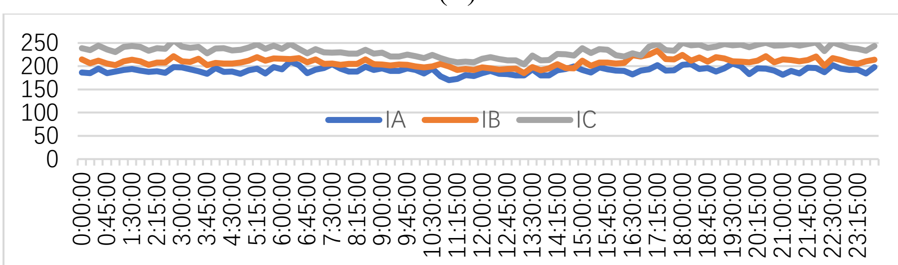
First, we analyze the current raw data. The following Fig. 8 is the data of IA, IB, and IC three-phase currents of five cities on 2017/1/27. It can be seen from Fig. 8 that some cities have similar data trend graphs of IA, IB, and IC at certain times. In this case, the three-phase unbalance may meet the specified requirements, and it is considered that the city is three-phase balanced on this day, as shown in Fig. 8(E). In some cities, the IA, IB, and IC three-phase data differ greatly at certain times. In this case, the three-phase unbalance may not meet the specified requirements, and it is considered that the city is unbalanced on this day. As shown in Figs. 8(A)-8(D). Therefore, if the three-phase unbalance can be predicted in advance, it will help the business department to adjust the balance in time and formulate targeted solutions to ensure a stable operation of the power grid.



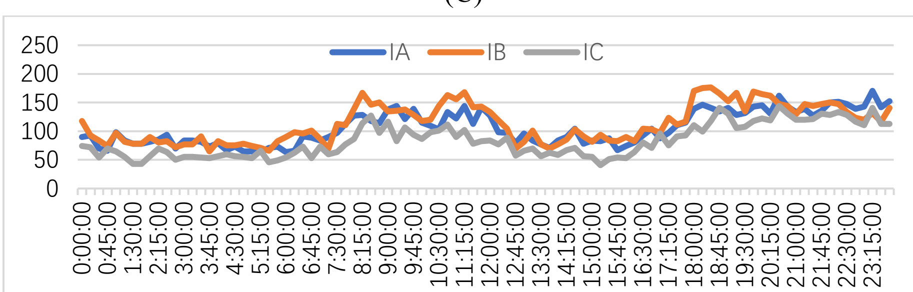
(A)



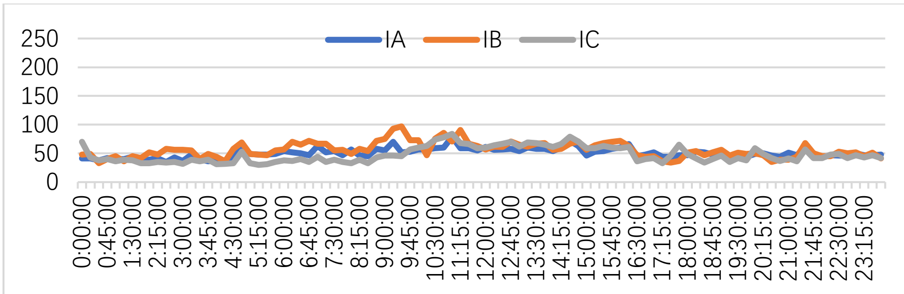
(B)



(C)



(D)



(E)

Figure 8: Three-phase current data on the 2017/1/27 in five cities (Tangshan, Langfang, Qinhuagndao, Chengde, Zhangjiakou)

4.2 Three-phase current unbalance analysis

Due to the difference in the amount of electricity used by each user and the difference in power usage time, the unbalanced current between the three phases in the power grid is objectively present, and there is no regularity in this power unbalance. For the three-phase unbalanced current, the power sector has almost no effective solution except to distribute the current as reasonably as possible.

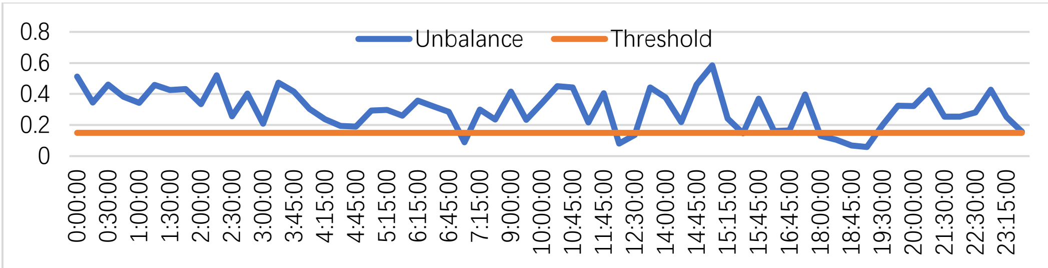
There are two concepts to explain here. Three-phase unbalance: It means that the three-phase current (or voltage) amplitude is inconsistent in the power system, and the amplitude difference exceeds the specified range. Unbalance: refers to the degree of three-phase imbalance in a three-phase power system.

According to the electricity inspection regulations [National Power Grid Corp Human Resources Department (2010)]: The three-phase current stress of the press should be balanced, and the unbalance should not exceed 0.15. When the above requirements are not met, the load (current) should be adjusted. The formula for calculating the imbalance is:

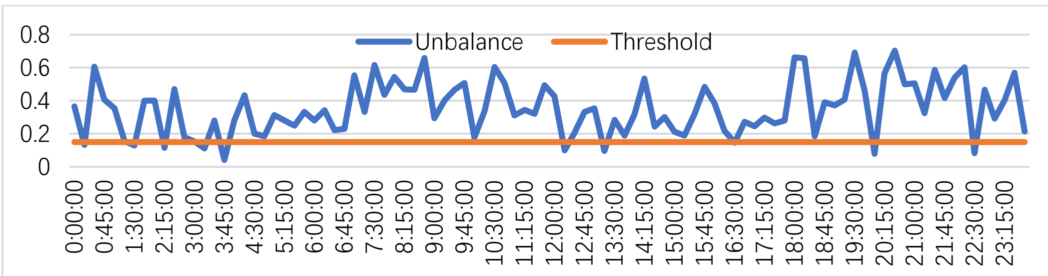
$$unbalance = \frac{max - min}{max} \tag{4}$$

Considering 96 points a day, if there is a point higher than the unbalance value, it is determined that the day is a three-phase unbalance. This condition is too strict. We add the definition of duration based on the above formula. That is, the total duration of the three-phase unbalance in one day is higher than the standard threshold is 288 minutes, or the continuous duration is 45 minutes, then it is determined that the city is a three-phase unbalance.

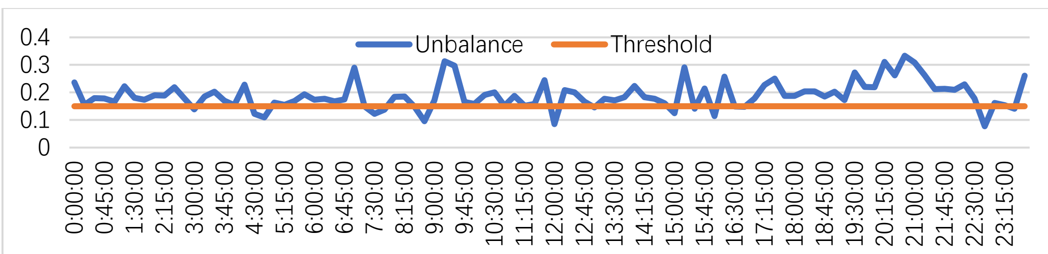
According to the data distribution of IA, IB and IC three phases on the same day in the five cities, the three-phase unbalance curve of the city on a particular day can be obtained using the Eq. (4) defined, plus the standard threshold in the regulation, and the standard threshold in the regulation can be drawn. A line chart showing the three-phase unbalance and the standard threshold. According to the analysis, it is known whether the city is unbalanced on a certain day (Fig. 9).


0.8
0.6
0.4
0.2
0
Unbalance
Threshold
0:00:00
0:30:00
1:00:00
1:30:00
2:00:00
2:30:00
3:00:00
3:45:00
4:15:00
4:45:00
5:15:00
6:15:00
6:45:00
7:15:00
9:00:00
10:00:00
10:45:00
11:45:00
12:30:00
14:00:00
14:45:00
15:15:00
15:45:00
16:45:00
18:00:00
18:45:00
19:30:00
20:00:00
21:30:00
22:30:00
23:15:00

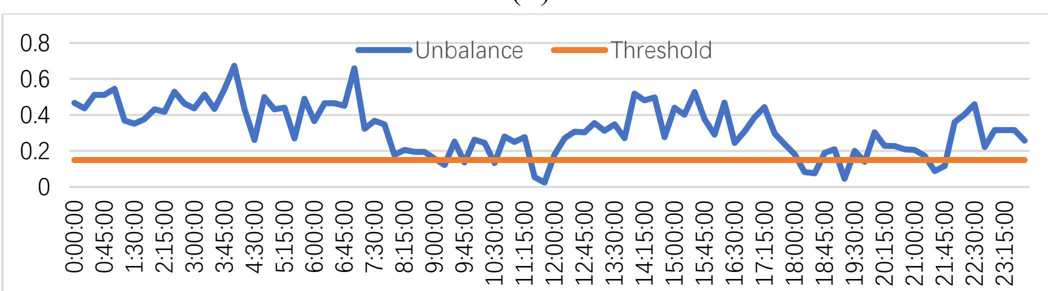
(A)


0.8
0.6
0.4
0.2
0
Unbalance
Threshold
0:00:00
0:45:00
1:30:00
2:15:00
3:00:00
3:45:00
4:30:00
5:15:00
6:00:00
6:45:00
7:30:00
8:15:00
9:00:00
9:45:00
10:30:00
11:15:00
12:00:00
12:45:00
13:30:00
14:15:00
15:00:00
15:45:00
16:30:00
17:15:00
18:00:00
18:45:00
19:30:00
20:15:00
21:00:00
21:45:00
22:30:00
23:15:00

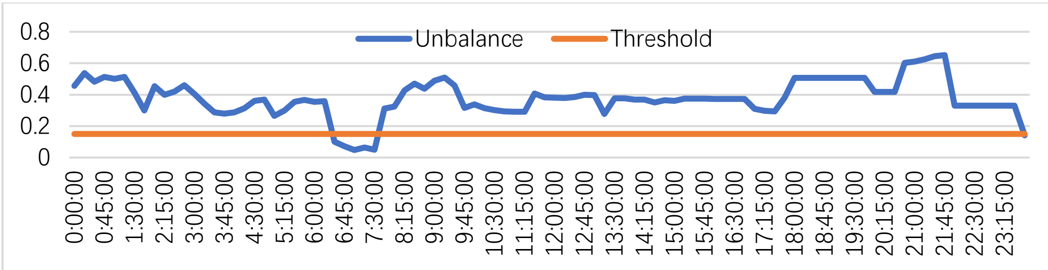
(B)


0.4
0.3
0.2
0.1
0
Unbalance
Threshold
0:00:00
0:45:00
1:30:00
2:15:00
3:00:00
3:45:00
4:30:00
5:15:00
6:00:00
6:45:00
7:30:00
8:15:00
9:00:00
9:45:00
10:30:00
11:15:00
12:00:00
12:45:00
13:30:00
14:15:00
15:00:00
15:45:00
16:30:00
17:15:00
18:00:00
18:45:00
19:30:00
20:15:00
21:00:00
21:45:00
22:30:00
23:15:00

(C)


0.8
0.6
0.4
0.2
0
Unbalance
Threshold
0:00:00
0:45:00
1:30:00
2:15:00
3:00:00
3:45:00
4:30:00
5:15:00
6:00:00
6:45:00
7:30:00
8:15:00
9:00:00
9:45:00
10:30:00
11:15:00
12:00:00
12:45:00
13:30:00
14:15:00
15:00:00
15:45:00
16:30:00
17:15:00
18:00:00
18:45:00
19:30:00
20:15:00
21:00:00
21:45:00
22:30:00
23:15:00

(D)



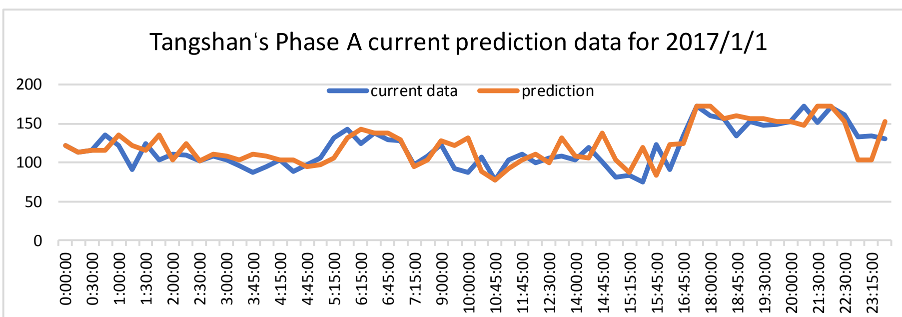
(E)

Figure 9: Three-phase unbalanced of five cities (Tangshan, Langfang, Qinhuagndao, Chengde, Zhangjiakou) on 2017/1/1

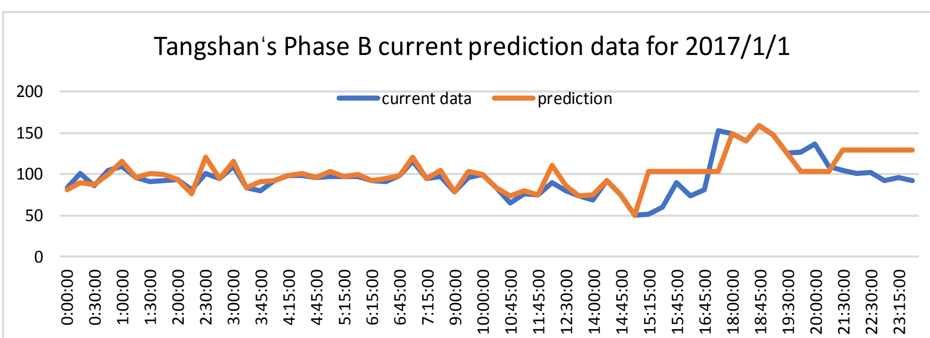
From the above analysis, the time of three-phase unbalance of current takes up most of the time of the day. Now, this situation is still dire, so it is necessary to use real-time prediction of HTM to distribute current in time and reasonably.

4.3 Current prediction

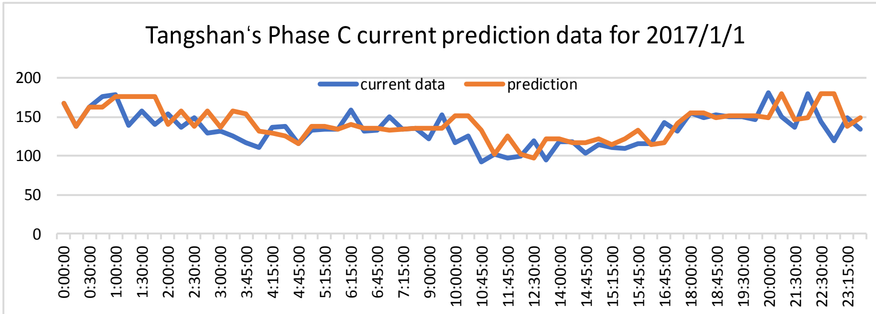
We use the HTM to predict the original data and show the prediction results and accuracy with Tangshan as Fig. 10 shown.



(A)



(B)



(C)

Figure 10: Three-phase current data prediction on the 2017/1/1 in Tangshan

The LSTM neural network has been improved based on the Recurrent Neural Network (RNN), which can solve the long-term dependence problem well. The effect of time-series data processing in the deep learning method is very significant. For the above reasons, we chose to compare the prediction accuracy of LSTM with HTM, as shown in the Tab. 3. In the table, the bold numbers indicate that the prediction result of HTM is better than LSTM. By comparison, HTM predictions are more accurate and stable in general, and HTM has a clear advantage in network structure, which will be more scalable in the future.

Table 3: Prediction accuracy of current data

City	Phase	HTM Accuracy	LSTM Accuracy
	IA	82.0%	84.0%
Tangshan	IB	84.3%	84.5%
	IC	**84.7%**	84.5%
	IA	**84.8%**	84.8%
Langfang	IB	**85.5%**	85.1%
	IC	**85.5%**	84.3%
	IA	**81.6%**	81.3%
Qinhuangdao	IB	**81.8%**	73.6%
	IC	81.4%	82.5%
	IA	85.0%	86.2%
Chengde	IB	**85.3%**	85.1%
	IC	**84.2%**	83.0%
	IA	**85.7%**	85.7%
Zhangjiakou	IB	81.5%	85.4%
	IC	84.0%	86.3%

5 Conclusion

In this paper, we elaborated on the construction of the HTM model and applied it to analyze the three-phase unbalance problem, which will seriously jeopardize the safe operation of the electrical equipment. It mainly predicts the three-phase currents of five cities and analyzes the three-phase unbalance in these cities based on the prediction results. And it is also compared with the long short-term memory (LSTM) algorithm in the experimental part, in which the predictions are more accurate and stable in general. Premature warning and monitoring of three-phase unbalanced can be achieved in this way. Through these warnings, the stable operation of the power grid can be ensured, the loss of transformers can be reduced, the safety index of transformers can be improved, and high-quality power services can be provided.

Funding Statement: This study is supported by the National Natural Science Foundation of China (No. 61801019).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Assaf, A.; Tammy, R. R. (2019): Microscopy cell segmentation via convolutional LSTM networks. *IEEE 16th International Symposium on Biomedical Imaging*, pp. 1008-1012.

Barth, A. L.; Poulet, J. F. (2012): Experimental evidence for sparse firing in the neocortex. *Trends in Neurosciences*, vol. 35, no. 6, pp. 345-355.

Boone, A. R.; Karnowski, T.; Chaum, E.; Giancardo, L.; Li, Y. et al. (2010): Image processing and hierarchical temporal memories for automated retina analysis, *Biomedical Sciences and Engineering Conference*, pp. 1-4.

Cui, Y.; Ahmad, S.; Hawkins, J. (2016): Continuous online sequence learning with an unsupervised neural network model. *Neural Computation*, vol. 28, no. 11, pp. 1-6.

Cui, Y.; Ahmad, S.; Hawkins, J. (2017): The HTM spatial pooler: a neocortical algorithm for online sparse distributed coding. *Frontiers in Computational Neuroscience*, vol. 11, pp. 111-115.

Fazle, K.; Somshubra, M.; Houshang, D.; Samuel, H. (2019): Multivariate LSTM-FCNs for time series classification. *Neural Networks*, vol. 116, pp. 237-245.

Hawkins, J. (2016): Biological and machine intelligence. Release 0.4.

Hawkins, J.; Ahmad, S. (2016): Why neurons have thousands of synapses, a theory of sequence memory in neocortex. *Frontiers in Neural Circuits*, vol. 10, pp. 23-25.

Hawkins, J.; Ahmad, S.; Dubinsky, D. (2010): *Hierarchical Temporal Memory Including HTM Cortical Learning Algorithms [Online Technical Report]*. Palo Alto, CA: Numenta, Inc.

Hebb, D. O. (1988): *The Organization of Behavior, Neurocomputing: Foundations of Research*. The MIT Press, USA.

Hochreiter, S.; Schmidhuber, J. (1997): Long short-term memory. *Neural Computation*, vol. 9, no. 8, pp. 1735-1780.

http://numenta.com/biological-and-machine-intelligence/.

http://www.numenta.com/assets/pdf/whitepapers/hierarchical-temporal-memory-cortical-learning-algorithm-0.2.1-en.pdf.

https://numenta.com/assets/pdf/apps/htmforstocks.pdf.

https://numenta.com/assets/pdf/apps/htmforstocks.pdf.

https://numenta.com/assets/pdf/biological-and-machine-intelligence/BaMI-Encoders.pdf.

Kirtay, M.; Falotico, E.; Ambrosano, A.; Albanese, U.; Vannucci, L. et al. (2016): Visual target sequence prediction via hierarchical temporal memory implemented on the iCub robot. *5th International Conference on Biomimetic and Biohybrid Systems*, vol. 9793, pp. 119-130.

Kouba, N. E. Y.; Menaa, M.; Tehrani, K.; Boudour, M. (2019): Optimal tuning for load frequency control using ant lion algorithm in multi-area interconnected power system. *Intelligent Automation and Soft Computing*, vol. 25, no. 2, pp. 279-294.

Liu, W.; He, J. L.; Li, M.; Jin, R.; Hu, J. J. et al. (2019): An efficient supervised energy disaggregation scheme for power service in smart grid. *Intelligent Automation and Soft Computing*, vol. 25, no. 3, pp. 585-593.

Major, G.; Larkum, M. E.; Schiller, J. (2013): Active properties of neocortical pyramidal neuron dendrites. *Annual Review of Neuroscience*, vol. 36, no. 1, pp. 1-5.

Mohamed, A. N.; Karar, M. (2019): Accurate photovoltaic power forecasting models using deep LSTM-RNN. *Neural Computing and Applications*, vol. 31, pp. 2727-2740.

National Power Grid Corp Human Resources Department (2010): *Electricity Inspection Regulations.* China Electric Power Press.

Numenta (2011): Detecting anomalies in publicly traded stocks using trading and twitter data.

Numenta (2011): Detecting anomalies in publicly traded stocks using trading and twitter data.

Numenta (2011): Geospatial tracking-learning the patterns in movement and detecting anomalies. https://numenta.com/assets/pdf/whitepapers/GeospatialTrackingWhitePaper.pdf.

Numenta (2011): Geospatial tracking—learning the patterns in movement and detecting anomalies. https://numenta.com/assets/pdf/whitepapers/GeospatialTrackingWhitePaper.pdf.

Numenta (2011): Hierarchical temporal memory including HTM cortical learning algorithms. https://numenta.org/resources/HTM_CorticalLearningAlgorithms.pdf.

Otahal, M.; Stepankova, O. (2014): Anomaly detection with cortical learning algorithm for smart homes. https://doi.org/10.13140/2.1.2927.8405.

Poirazi, P.; Brannon, T.; Mel, B. W. (2003): Pyramidal neuron as two-layer neural network. *Neuron*, vol. 37, no. 6, pp. 989-999.

Polsky, A.; Mel, B. W.; Schiller, J. (2004): Computational subunits in thin dendrites of pyramidal cells. *Nature Neuroscience*, vol. 7, no. 6, pp. 621-627.

Purdy, S. (2016): Encoding data for HTM systems.

Spruston, N. (2008): Pyramidal neurons: dendritic structure and synaptic integration. *Nature Reviews Neuroscience*, vol. 9, no. 3, pp. 206-221.

Wang, W. (2013): *Research on Individual Memory-Prediction Model of Rumor Propagation Based on HTM (Ph.D. Thesis)*. Huazhong University of Science and Technology.

Wu, J.; Zeng, W.; Chen, Z.; Tang, X. F. (2016): Hierarchical temporal memory method for time-series-based anomaly detection. *16th IEEE International Conference on Data Mining Workshops*, pp. 1167-1172.

Yu, R. G.; Liu, Z. Q.; Wang, J. R.; Zhao, M. K.; Gao, J. et al. (2018): Analysis and application of the spatio-temporal feature in wind power prediction. *Computer Systems Science and Engineering*, vol. 33, no. 4, pp. 267-274.

Zhou, Y. J. (2018): *Research on Power Load Three-Phase Unbalanced Prediction Model Based on Machine Learning (Master Thesis)*. University of Science & Technology Beijing.

Zhu, Q. D.; Yang, Z. B. (2019): Intelligent power compensation system based on adaptive sliding mode control using soft computing and automation. *Computer Systems Science and Engineering*, vol. 34, no. 4, pp. 179-189.