



**Michigan
Technological
University**

Michigan Technological University
Digital Commons @ Michigan Tech

Michigan Tech Publications

5-4-2022

A Few-shot Learning Model based on a Triplet Network for the Prediction of Energy Coincident Peak Days

Jinxiang Liu

Michigan Technological University, jliu15@mtu.edu

Laura Brown

Michigan Technological University, lebrown@mtu.edu

Follow this and additional works at: <https://digitalcommons.mtu.edu/michigantech-p>



Part of the [Computer Sciences Commons](#)

Recommended Citation

Liu, J., & Brown, L. (2022). A Few-shot Learning Model based on a Triplet Network for the Prediction of Energy Coincident Peak Days. *Proceedings of the International Florida Artificial Intelligence Research Society Conference, FLAIRS, 35*. <http://doi.org/10.32473/flairs.v35i.130733>
Retrieved from: <https://digitalcommons.mtu.edu/michigantech-p/16164>

Follow this and additional works at: <https://digitalcommons.mtu.edu/michigantech-p>



Part of the [Computer Sciences Commons](#)

A Few-shot Learning Model based on a Triplet Network for the Prediction of Energy Coincident Peak Days

Jinxiang Liu and Laura E. Brown

Michigan Technological University
1400 Townsend Drive
Houghton, MI 49931 USA

Abstract

In an electricity system, a coincident peak (CP) is defined as the highest daily power demand in a year, which plays an important role in keeping the balance between power supply and its demand. Advanced information about the time of coincident peaks would be helpful for both utility companies and their customers. This work addresses the prediction of the five coincident peak days (5CP) in a year. We present a few-shot learning model to classify a day as a 5CP day or a non-5CP day 24-hours ahead. A triplet network is implemented for the 2-way-5-shot classifications on six different historical datasets. The prediction results have an average (across the six datasets) mean recall of 0.933, mean precision of 0.603, and mean F1 score of 0.733.

Introduction

Keeping an electricity system stable requires that the supply of power is always sufficient to cover the demands. To stimulate customers to lower the electricity usage at peak times, many regional transmission organizations or independent system operators implement pricing programs that require large customers to pay surcharges during the time of coincident peaks (days of highest power demands). For example, in Ontario, Canada, the Independent Electricity System Operator (IESO) charges their customers a higher price on five coincident peaks (5CP), which are the top five daily maximum power demands in a year. The system operators send out notifications of predictions on future demands, but the timing of whether a day is a CP is not confirmed until after the fact (end of the fiscal year for IESO). Therefore, accurate predictions of coincident peaks may be beneficial to both system operators and consumers.

Our work is for the day-ahead prediction of the five coincident peaks (5CP days). Because there are only five CP days in a year ($\sim 1.4\%$), the prediction of 5CP days is an imbalanced classification problem. In order to obtain accurate results, most learning models need to be trained on sufficient data, in some problems the available data is limited. Under such circumstances, few-shot learning may be a solution, as it is capable of learning from a small amount of data. Unlike other classification methods, few-shot learning models learn how to learn. It gains experience from a set of problems rather than some particular problem.

Because of the insufficient positive cases (5CP days), few-shot learning models may help in predicting the five coincident peak days. Therefore, a few-shot learning model based on a triplet network is developed to predict whether the next day is a 5CP day or not. Actual historical data from six different areas is used to train and test the model. Note, since a model in practice would have forecasted data also available for its use, the true data is also used as forecasted data from an oracle; thus, the results are optimistic but can be used to understand the feasibility of the approach. Through the leave-one-out cross-validation scheme by year, the prediction results indicate this model performs well on this problem with the average (across six datasets) mean recall of 0.933, mean precision of 0.603, and mean F1 score of 0.733.

Related Work and Background

There is a tremendous amount of literature on forecasting electricity demands (Kandil, El-Debeiky, and Hasanien 2002; Li and Li 2017). Many different approaches have been applied to this problem from naive methods, e.g., autocorrelation with weather data (McSharry, Bouwman, and Bloemhof 2005), autoregressive integrated moving average (Li and Li 2017), to machine learning models, such as K-nearest neighbors (El-Attar, Goulernas, and Wu 2009), fuzzy models (Ying and Pan 2008), and artificial neural networks (Saini 2008). However, the prediction of the day of five coincident peaks (5CP) is not like the prediction of power demands (a regression problem). The prediction of 5CP days needs to locate the date of the 5CP in a year. Because of the imbalanced classes, it could be hard to predict 5CP days accurately through traditional prediction models.

In Jiang et al., a heuristic algorithm is proposed to predict whether the next day will be the day with one of highest five power demands in a fiscal year in IESO (Jiang et al. 2014). Relying on the fact that residuals of power demand forecasts from 2006 to 2013 are normally distributed and the presumption that power demand forecasts are independent for different days, they used the 14-day short-term power demand forecasts and weather forecasts for the next day to calculate the probability that one of 5CP occurs tomorrow, and then set the probability threshold derived from the estimates of previous years to classify 5CP days. The forecast data is no longer available therefore, our analysis does not take it into account. The work from Ryu et al. is about forecasting

the time of five coincident peaks in IESO (Ryu et al. 2016). Naïve Bayes was used to predict daily peak, 3-hour peak and 1-hour peak for 5CP. Their model has been trained and tested on IESO hourly data over 21 years. Different from the work from Jiang et al., the probability threshold was set as 0.5, a fixed number, instead of a heuristic one. In addition, their analysis over expanded years 1995-2015 had 5CPs occurring during both the summer and winter.

Background

Unlike most learning models, which require abundant data to train the model, few-shot learning can gain experience from few examples (Wang et al. 2020). Because it is learning how to learn from a set of tasks, few-shot learning can be characterized as an example of meta-learning.

Generally, few-shot learning can be considered as a M -way- N -shot classification, where M indicates the number of classes, and N means the number of examples per class. The training and testing for the few-shot learning model consist of a sequence of tasks. Each task has a support set and a query set. The support set teaches the model how to solve the task, while the query set assesses the model performance on the task. The support set contains $M \times N$ labeled examples with M classes and N examples per class. The classes in the support set could be different in different tasks. The query examples should be different from the support examples, but they have the same classes. After learning from the different training tasks, the model should have the ability to discriminate classes. During the training, the cost on the query set of the task assesses the model performance.

Testing tasks are for model testing and should have different examples from the training tasks. The testing task also has a support set and a query set. For the testing task, the support examples have the same classes with the query example. According to the given support set in the testing task, the model classifies the examples in the query set, and their classification results indicate the performance of the model.

Triplet network We implement few-shot learning by using a triplet network, but many other few-shot learning models could be chosen, for example, prototypical networks or matching networks (Snell, Swersky, and Zemel 2017). Triplet networks (Hoffer and Ailon 2015) learn similarity and dissimilarity among the samples. It takes triplets as inputs, consisting of three examples: anchor, positive and negative examples (Ach., Pos. and Neg.). The anchor example is selected to describe some class, and it could be any arbitrary sample in the dataset. The positive example should be any other sample from the class of anchor example, while any sample from a different class can be the negative example. During the training, three examples in the triplet will be passed through three identical, base neural networks. The base networks share weights with each other. Through the base networks, three embeddings (E_A , E_P , and E_N) would be generated and then will be used to calculate the distances in the embedding space between the anchor and positive examples ($D+$), and between the anchor and negative examples ($D-$). Triplet network uses triplet loss as the learning criterion, which pushes the anchor example to be distant

from the negative example, but close to the positive example. After training, the positive example should have a shorter distance to the anchor example than the negative example.

In the testing phase, the model takes two examples. One example (Tag.) is the testing case, and its class is unknown, while the class of the other example (Sup.) is already known. These two examples will be passed through the trained base networks to generate embeddings (E_T and E_S). The distance between these two embeddings (D) determines whether the testing case has the same class as the other example or not.

To implement the few-shot learning technique, the triplet network needs to be mapped to the few-shot learning framework. For the model training, the anchor example is selected from the query set of the training task, while the positive and negative examples are from the support set but they have different classes. For the model testing, the class of the example in the query set is determined by the distance to the examples in the support set.

Triplet loss Triplet loss (Schroff, Kalenichenko, and Philbin 2015) is a loss function that takes the embedding for the anchor, positive and negative examples. Triplet loss is defined as:

$$Loss(E_P, E_A, E_N) = \max(0, \alpha + d(E_A, E_P) - d(E_A, E_N)),$$

where α represents the margin of the positive examples and negative examples in the embedding space, E_A is the embedding for anchor example, E_P is the embedding for positive example, and E_N is the embedding for negative example. Function $d(A, B)$ calculates A 's distance from B . In our work, the distance between the two embeddings are measured by squared Euclidean distance.

According to the location of the negative example in the embedding space, relatively to the positive and anchor examples, there are three kinds of triplets:

- **Easy triplets**, where $d(E_A, E_P) + \alpha < d(E_A, E_N)$: The triplet loss has a value of 0. In the embedding space, the negative example is far enough away from the anchor example concerning the positive example.
- **Semi-hard triplets**, where $d(E_A, E_P) < d(E_A, E_N) < d(E_A, E_P) + \alpha$. The triplet loss has a positive value smaller than the margin α . In the embedding space, the negative example has a larger distance to the anchor example than the positive example, however, the distance difference is less than α .
- **Hard triplets**, where $d(E_A, E_N) < d(E_A, E_P)$: Triplet loss has a positive value greater than the margin α . The negative example has a shorter distance to the anchor example than the positive example in the embedding space.

Approach to Predict Coincident Peak Days

This work looks to predict whether tomorrow is a five coincident-peak day. Like the approach presented by (Ryu et al. 2016), we used classification models to predict 5CP days. Attributes include historical power demands and weather, dates, as well as power demand and weather forecasts. We developed a few-shot learning model to classify the next day into 5CP days or non-5CP days.

Table 1: Data set information

ISO/RTO	Area	Weather Station	Date Range
IESO	Ontario	CYKZ	21 yrs; 5/1/94-4/30/15
PJM	DPL	ILG	24 yrs; 1/1/94-12/31/17
PJM	PS	EWR	24 yrs; 1/1/94-12/31/17
MISO	LRZ1	MSP	3 yrs; 6/1/15-5/31/18
MISO	LRZ2-7	MKE	3 yrs; 6/1/15-5/31/18
MISO	LRZ4	SPI	3 yrs; 6/1/15-5/31/18

Data

The hourly data used in our work includes power demands and weather from six different areas. The power demand data is provided by Independent Electricity System Operator, IESO (IESO 2022), PJM a regional transmission operator in 13 states (PJM 2022), and Midcontinent Independent System Operator, MISO (MISO 2022). The corresponding weather data comes from the Canadian government climate website¹ (for IESO) and NOAA’s National Centers for Environmental Information² (for the other five areas).

Information on the datasets is summarized in Table 1. Two zones for PJM are studied in our work, Public Service Electric & Gas zone (PS) and Delmarva Power & Light zone (DPL). In addition, we examined three local resource zones (LRZs) from MISO, which are LRZ1, LRZ4, and LRZ2-7 (it combines LRZ2 with LRZ7 together).

For each weather station (airport code in Table 1), the air temperature, relative humidity, humidex and windchill are taken into consideration. Humidex and windchill are calculated by using air temperature, dew point temperature and wind speed. Humidex is not displayed when the air temperature is less than 20°C or the value of humidex is more than 1°C less than the air temperature, and windchill does not exist when the wind speed is 0 km/h or the air temperature is over 0°C. For such cases, the value of humidex and windchill has been set as -1 and 1 respectively. Any missing values in the datasets were replaced using linear interpolation.

In our few-shot learning model, the historical power demands, calendar information, and the next day forecasts of power demands and weather factors have been considered. Note, daily forecast data is not available, therefore the actual data is used (forecasts from an oracle). Thus, our results are optimistic, yet important in establishing the promise of this approach. Table 2 lists the discrete and continuous attributes. We used the technique of binary encoding to convert discrete attributes to binary codes.

Implementation of triplet network

A triplet network has been selected to identify whether a day is a 5CP day or a non-5CP day. The model takes three examples as the input, which are anchor, positive and negative examples. We used a random number generator to determine the class of the anchor example. If the number generated by the generator is 1, the anchor example should come

¹Canada: Historical Climate Data, <http://climate.weather.gc.ca/>

²NOAA, <https://www.ncdc.noaa.gov/cdo-web/>

Table 2: Attributes for modeling. The subscript ph indicates values reported at the peak power demand hour.

Discrete Attributes	Description
Month (Jan. to Dec.)	M_1, \dots, M_{12}
Day of week (Mon. to Sun.)	W_1, \dots, W_7
Holiday indicator (working day vs. holiday/weekend)	$H = \{0, 1\}$
Continuous Attributes	Description
Humidex at peak hour (ph)	H_{ph_0}
Temperature at ph	T_{ph_0}
Windchill at ph	WC_{ph_0}
Relative humidity at ph	RH_{ph_0}
Power demand at ph	P_{ph_0}
Historical power demand peaks, prior two weeks	$P_{ph-14}, P_{ph-13}, \dots, P_{ph_0}$
Forecasts of hourly power demand	$FP_1, FP_2, \dots, FP_{24}$

from the class of 5CP days, otherwise, non-5CP days. Then two different cases are randomly chosen from the decided anchor example class as the anchor and positive examples, whereas the negative sample is a day randomly selected from the other class.

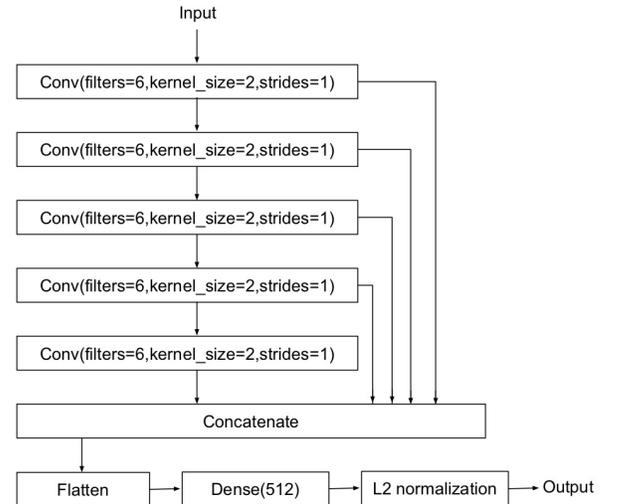


Figure 1: Architecture of the base network

Figure 1 exhibits the architecture of the base network we used within the triplet network. It is a convolutional neural network with five one-dimensional convolutional layers in a sequence. Each convolutional layer has 6 filters with a size of 2 and a stride with the value of 1. After that, there is a fully connected layer, which takes the outputs of convolutional layers (after they are concatenated and flattened) as the inputs. This layer has 512 units and no activation. To normalize the representations, a normalization layer has been added after the fully connected layer. The output of the base network is called the embedding.

During the triplet network training, the base network

maps three selected examples (or a triplet) to the embedding space. Then in the embedding space, we calculated the squared Euclidean distances between the anchor and positive examples ($D+$), as well as between the anchor and negative examples ($D-$). As the learning criterion, a triplet loss function with the margin α of 0.5 was used to train the model. Since triplet loss encourages the value of $D+$ to be smaller than the value of $D-$, then after the training, the negative example should be farther from the anchor example than the positive example.

Because the triplet loss for easy triplets equals 0, we have to avoid using easy triplets to train the triplet network. To achieve that, only 60-CP days for each fiscal year are kept in the training dataset (60-CP days are the 60 days with the highest daily load peak in a fiscal year).

During the testing of the triplet network, two examples are passed through the base network to map them to the embedding space. The class of one example is already known, while the other example is one of the testing cases with an unknown class. The squared Euclidean distance between these two examples in the embedding space (D) is calculated, which will be used to determine whether they are from the same class or not.

Few-shot learning model based on a triplet network

A 2-way-5-shot learning model based on a triplet network is implemented to address our problem. The few-shot learning models are trained and tested on a series of tasks. We first need to form the tasks from the training and testing data.

There are a support set and a query set for each task (a training task or a testing task). The model is learned from the support set while it is evaluated on the query set. The support set should contain 10 examples with 5 different examples per class, namely, five 5CP days and five non-5CP days. In our work, the query set has five examples for model training, but only one example for model validation or testing. The query example could be either a 5CP day or a non-5CP day. Suppose the five 5CP days in the support set are Dp_1, \dots, Dp_5 , the five non-5CP days in the support set are Dn_1, \dots, Dn_5 , and Dt (for testing) or Dt_1, \dots, Dt_5 (for training) represent the days in the query set.

For every training task, the class of the example in the query set is determined by the random number generator. Then, the support and query examples are randomly selected from the training data (ensuring they are separate examples).

To train the triplet network, the input triplets are created by using the examples in the training task. The example in the query set is set as the anchor example of the input triplet, while the positive and negative examples come from the support set. For example, if the example in the query set, Dt_1 , has the class of 5CP days, one input triplet could be Dt_1 as the anchor example, Dp_1 as the positive example, and Dn_1 as the negative example. In our work, five different triplets are generated from one training task, which are $\{Dt_1, Dp_1, Dn_1\}, \dots, \{Dt_5, Dp_5, Dn_5\}$. Figure 2 illustrates the process that converts the training task to the input triplets for model training.

During model training, new training tasks with only one query example are used to validate the model performance.

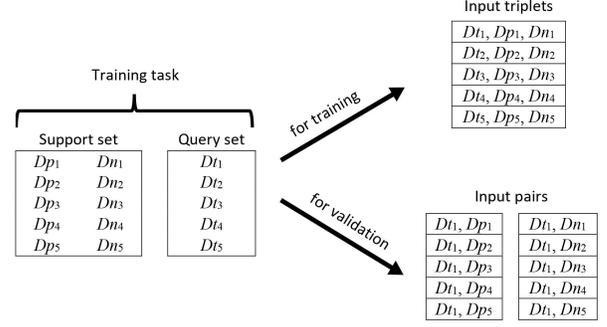


Figure 2: Training task to the input of triplet network

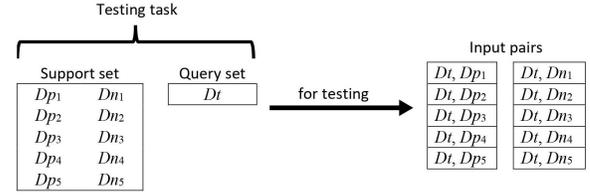


Figure 3: Testing task to the input of triplet network

The input to the triplet network consists of two examples with one query example and one support example. The query example works as the example with the class unknown while the support example is the example with the class labeled. To make sure that the model can classify actual 5CP days effectively during the model validation, the example in the query set is always selected from the class of 5CP days. The following four steps are for model validation or testing, and Figure 4 exhibits such an example.

Step 1. Pair every support example with the query example to form the input to the triplet network. In total, it should have 10 input pairs, since there are 10 examples in the support set, see Fig. 2 and Fig. 3.

Step 2. Pass the input pairs through the triplet network to obtain the distance between two examples in the embedding space. Let Op_1, \dots, Op_5 represent the outputs for the input pairs with the support example being a 5CP day, and On_1, \dots, On_5 be the outputs for input pairs with the support example being a non-5CP day.

Step 3. Calculate the averages, Ap and An of $\{Op_1, \dots, Op_5\}$ and $\{On_1, \dots, On_5\}$ respectively.

Step 4. Decide the class of the example in the query set. If $Ap < An$, then the example is believed to be a 5CP day. If $Ap > An$, then this example is classified as a non-5CP day. If $Ap = An$, the validation will be repeated on the task with different support set but same query set (this case happens with a very small probability).

In our work, we validated the performance of the model after it has been trained on 300 different training tasks. Because the classification results highly depend on the selected

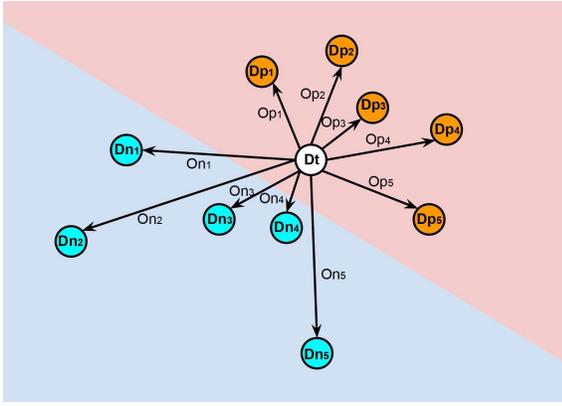


Figure 4: An example for model validation or testing

support examples, to reduce the variance caused by the selection, 200 different training tasks are used for the model validation. After the accuracy of the prediction reached 100%, or the model has been trained on 42,000 training tasks, we stopped the training of the model.

For each testing task, the support examples are also randomly chosen from the training data. However, the example in the query set is from the testing data. This example could be either a 5CP or non-5CP day, and its class is unknown. Like how the model is validated during the training, the same steps are used for model testing on testing tasks. The class of the query example is also decided by the class with a smaller average of distance. To reduce the variance of the classification results, we choose to use 51 testing tasks with different support sets but the same query set to predict whether the query example is a 5CP day or not (another number of testing tasks also work, but the number has to be odd and large enough to reduce the variance). If more than 25 tasks indicate this example is from a given class, we can believe that day is the same class.

Results

Metrics including precision, recall, F1 score and LRank have been used to evaluate the model performance. The number of true positives (TP), false positives (FP), and false negatives (FN), are used to calculate precision, $precision = \frac{\#TP}{\#TP + \#FP}$, and recall, $recall = \frac{\#TP}{\#TP + \#FN}$. F1 score is calculated as the harmonic mean of recall and precision.

We define a new evaluation metric, LRank, which measures the largest actual demand ranking of predicted 5CP days. At the end of every fiscal year, each day in the year is sorted and ranked with descending order based on the highest power demand for that day. LRank reports the maximum ranking of the days that are predicted as 5CP days.

The experiment has been run five times for every dataset, with the average of these five runs reported. We also implemented the scheme of leave-one-out cross-validation by years (e.g., if the dataset contains 21-year data, then the training dataset consists of the data for 20 years, and the remaining 1-year data is for the testing dataset).

Due to the lack of the historical datasets of hourly forecasts for power demands and weather condition, the actual values (oracle data) for the next day have been used to test the model. In addition, considering the limited size of positive data, the data has been min-max normalized by year during the data preprocessing to reduce the yearly variance, then the data for each year could be more independent with less temporal impact on trends from other years. Usually, it is impossible to know the actual minimum and maximum values for each attribute over a fiscal year at the beginning of the year. The minimum and maximum values for each attribute could be predicted accurately by studying its changes and distributions over the years, and it will be a part of our future work. Therefore our prediction results are an *optimistic view* of the problem.

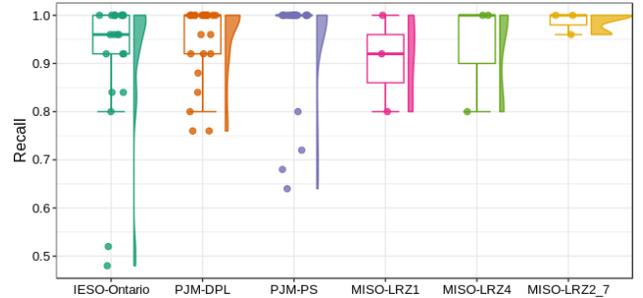


Figure 5: Recall results over six datasets

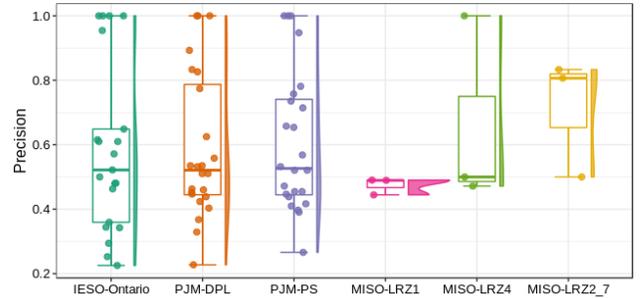


Figure 6: Precision results over six datasets

Figure 5 exhibits the result of recall for each dataset based on the averages over five experiments. Overall, the few-shot learning model is able to identify the 5CP days. For IESO-Ontario, there are only 2 years the value is less than 0.80. Similarly, only 2 years for PJM-DPL and 3 years for PJM-PS is the recall less than 0.80. Figure 6 displays the distribution of precision on testing cases. The precision shows a much larger range of values for all data sets except for MISO-LRZ1.

Table 3 lists the average values of the number of true positives (# TP), the number of false positives (# FP), recall, precision (Prec.), F1 score (F1) and LRank for each dataset over the years. The recall averaged across all six datasets is 0.903, and the average of recall for MISO-LRZ2-7 is 0.99. The precision averaged across all six datasets is 0.603. MISO-

LRZ2-7 has the highest average precision with a value of 0.71. The F1 score averaged across all six datasets is 0.733. The average F1 scores range from 0.62 to 0.83. The LRank metric averaged across all six datasets is 33.5, with a median of 15. For five of the datasets, the LRank metric is between 9 and 21. Meaning of the days predicted to be 5CP days by the model, the maximal actual lowest rank of these days is in this range. The false positives predicted by the model are not too far away from the actual 5CP day (ranks 1-5). MISO-LRZ2-7 shows a much worse result, which bears further scrutiny. In summary, this few-shot learning model exhibits promising prediction results.

Table 3: Average performance over years for each metric

Area	# TP	# FP	Recall	Prec.	F1	LRank
IESO-Ontario	4.54	5.28	0.91	0.58	0.71	16.8
PJM-DPL	4.73	4.58	0.95	0.59	0.73	12.9
PJM-PS	4.76	4.22	0.95	0.61	0.74	21.0
MISO-LRZ1	4.53	5.00	0.91	0.47	0.62	13.1
MISO-LRZ4	4.67	3.53	0.93	0.66	0.77	9.7
MISO-LRZ2-7	4.93	2.33	0.99	0.71	0.83	127.5
Average	4.69	4.16	0.93	0.60	0.73	33.5

Conclusion

This work focuses on the prediction of 5CP days. A few-shot learning model based on a triplet network has been developed for this prediction problem. As the base network in the triplet network, a convolutional neural network maps the input examples to embedding space. A triplet loss with a margin of 0.5 has been used as the learning criterion to train the model. To prevent the model being trained on easy triplets, only the 60 days with the highest daily power peak in every fiscal year are kept to make the training tasks. A leave-one-out cross-validation scheme by year has been applied to train and test the model on six different sets of actual historical data. Examining the evaluation metrics on prediction results, this few-shot learning model performed well with the mean recall of 0.933, mean precision of 0.603, mean F1 score of 0.733, and mean LRank of 33.5 averaged over the six historical datasets.

Several directions could be considered for future work. First, feature selection technique could be incorporated into the model. It is possible to improve the model performance through using the best combination of the attributes indicated by the feature selection method. Second, techniques like generative adversarial network (GAN), meta-learning, and deep reinforcement learning would be implemented to address the prediction of 5CP days. Similar to few-shot learning, such approaches learn from a small amount of data actively. Finally, the model could be extended to predict the hour that five coincident peaks occur. As the prediction of 5CP hours works on more imbalanced dataset, it would be more challenging to address.

Acknowledgments

This work is supported in part by NSF GCR-1934346.

References

- El-Attar, E.; Goulermas, J.; and Wu, Q. 2009. Forecasting electric daily peak load based on local prediction. In *Proceedings of 2009 IEEE Power & Energy Society General Meeting*, 1–6.
- Hoffer, E., and Ailon, N. 2015. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, 84–92. Springer.
- IESO. 2022. <http://www.ieso.ca/power-data/data-directory>. Accessed: Jan. 29, 2022.
- Jiang, Y. H.; Levman, R.; Golab, L.; and Nathwani, J. 2014. Predicting peak-demand days in the ontario peak reduction program for large consumers. In *Proceedings of the 5th International Conference on Future Energy Systems, e-Energy '14*, 221–222. New York, NY, USA: ACM.
- Kandil, M.; El-Debeiky, S. M.; and Hasanien, N. 2002. Long-term load forecasting for fast developing utility using a knowledge-based expert system. *IEEE transactions on Power Systems* 17(2):491–496.
- Li, S., and Li, R. 2017. Comparison of forecasting energy consumption in shandong, china using the arima model, gm model, and arima-gm model. *Sustainability* 9(7):1181.
- McSharry, P. E.; Bouwman, S.; and Bloemhof, G. 2005. Probabilistic forecasts of the magnitude and timing of peak electricity demand. *IEEE Transactions on Power Systems* 20(2):1166–1172.
- MISO. 2022. MISO market and operations. <https://www.misoenergy.org/markets-and-operations/>. Accessed: Jan. 29, 2022.
- PJM. 2022. PJM market and operations. <https://www.pjm.com>. Accessed: Jan. 29, 2022.
- Ryu, B.; Makanju, T.; Lasek, A.; An, X.; and Cercone, N. 2016. A naive bayesian classification model for determining peak energy demand in ontario. In *Smart City 360*, 517–529.
- Saini, L. M. 2008. Peak load forecasting using bayesian regularization, resilient and adaptive backpropagation learning based artificial neural networks. *Electric Power Systems Research* 78(7):1302–1310.
- Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 815–823.
- Snell, J.; Swersky, K.; and Zemel, R. S. 2017. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*.
- Wang, Y.; Yao, Q.; Kwok, J. T.; and Ni, L. M. 2020. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)* 53(3):1–34.
- Ying, L.-C., and Pan, M.-C. 2008. Using adaptive network based fuzzy inference system to forecast regional electricity loads. *Energy Conversion and Management* 49(2):205–211.