An Implementation of Machine Learning for Parkinson's Disease Diagnosis

Hassapol Thummikarat
Department of Computer Engineering,
Chulalongkorn University,
Bangkok, Thailand.
e-mail : luping49030@hotmail.com

Prabhas Congstitvatana, Ph.D., Prof.
Department of Computer Engineering,
Chulalongkorn University,
Bangkok, Thailand.
e-mail : prabhas@chula.ac.th

***Abstract*— Machine learning is widely used in the medical applications. Parkinson's disease is a nervous system disorder which commonly causes tremors, but the disorder also commonly causes stiffness or slowing of movement. These symptoms are not only caused by Parkinson's disease but also the other movement disorder sickness. The doctors who are specialist in the Parkinson's disease can simply diagnose the tremors, which usually be hand muscle tremor of the patient. But conversely, the out-patient-department doctors find that it is difficult to diagnose those symptoms. This work proposes the use of machine learning for the Parkinson's disease data to assist the physician diagnosis. The Long Short-Term memory network is suitable for the data collected by a specialist. The result shows that the proposed method has 73% accuracy in early identifying the patient with Parkinson's disease.**

Keywords—Long Short-Term memory (LSTM), Parkinson's disease diagnosis, machine learning, neural networks , binary classification

I. INTRODUCTION

The Parkinson's disease (PD) is a brain and nervous disorder. Its symptoms are obviously tremor, stiffness, and slow movement. It is vital for patients to be diagnosed as early as possible, in order that the patients are treated since the early stage of the sickness to prolong their life. Outpatient department (OPD) doctors are facing the diagnosis problem of tremors which can be causing by PD or the other diseases, on the other hand, the PD specialists simply manage to judge the diagnosis of tremors. But it is not possible for PD doctors to be at the OPD, thus the OPD doctors require a tool to improve their diagnosis before sending all those patients with tremors to PD department. In order to assist the doctor, this work proposes Machine learning method to help with the diagnosis. There are two neural network models to be compared, the simple neuron model and the LSTMs model.

Machine learning algorithms have been widely used for medical data analysis. PD and heat disease data was used to classify the difference between these two diseases, as in [1]. An application of neural network with EEG, video and tracking data chose LSTM to solve the problem, as in [2]. To establish the strategy for the intensive care unit (ICU), Recurrent Neural networks (RNNs) and Long Short-Term Memory (LSTM) are used with Electronic Health Record (EHR) in the clinic measurement and management [3]. The classification method for EEG used LSTMs to classify the data from wearable devices in real-time [4]. Various disease data is time series data and LSTMs are used to cluster patients [5].

The LSTMs are an improvement of RNNs to overcome the vanishing gradient problem. RNNs have difficulties with time dependencies. LSTMs have advantage to overcome this problem, as in [6]. Time-series analysis, using deep learning shown the application of the NNs to time dependency information, as in [7], [8] and [9]. Time-series classification required the matching data sequence, as in [10]. LSTMs were used for imaging classification, as in [11]. Time-series classification could use the LSTMs for classifying the sequences, as in [12], [13] and [14].

The applications of the Parkinson's Disease data for diagnosis are interesting and challenging in both engineering and medical development over years. Using fuzzy method was an alternative approach to the PD diagnosis, as in [15]. The Extreme Leaning Machine (ELM) was used as a hybrid kernel ELM and its potential was sufficient, as in [16]. RNNs could predict PD, as in [17]. Deep Learning (DL) could be used to obtain the classifying and predicting the PD in case of large scale data, as in [18] and [19]. LSTMs were used to receive the PD subtypes from the clinical records, as in [20]. A simple ML method called Support Vector Machine (SVM) generating a solid outcome, as in [21]. Early diagnosis for PD patients used ML to predict from various data sources, as in [22]. Smartphone applications were introduced to apply AI, as in [23] and [24]. The NNs method was the best application option for this type of data ,as in [25].

This work describes an implementation of ML to improve the diagnosis of PD. The implementation consists of data preparing, neural network modelling, parameters tuning and experimenting the model.

II. IMPLEMETATION

A. Data Preparation

The raw data is contained of records from PD patients and non PD persons called control group, the records are from a keyboard and sensors collecting through a controller as shown in Fig. 1.

Figure 1. Tools for collecting PD data.

The keyboard data and the gyroscope and accelerometer sensors are collected at different frequency and have different features. Since the sensor has more sample rate than the keyboard, thus it is necessary to rescale and resample the keyboard data. The data is normalized as following.

- correct key : 1
- incorrect key : -1
- burst key : -0.5
- repeat key : 0.5
- double press : -0.25

These rules generate keyboard data which are convenient to analyze.

```
Algorithm for creating new keyboard data:
sensor data index = 0
keyboard data index = 0
new keyboard data = []
loop sensor index in range of sensor data length :
    check keyboard time >= sensor time:
        use the rules for keyboard feature data
        ++keyboard index
    else : use 0 as keyboard feature data
    ++sensor index
```

The above algorithm generates a new keyboard data, with frequency equal to the sensor data, and the data is normalized. At 40 Hz, the both data of keyboard and sensor are equal length.

For supervised learning, features and outputs forming pairs of input and output sequences. The data are organized into three sets as follows.

- Dataset-1 : keyboard dataset
- Dataset-2 : sensor dataset
- Dataset-3 : sensor and keyboard dataset

The input shape of LSTMs network must be three dimensions meaning which the datasets from data preparation process need to be reshape as input shape of 3D data (samples , time steps, features) for the LSTMs model or input dimension (features) for the simple neuron model.

All data of the 2D datasets are reshaped into 3D datasets as the (samples × time steps × features) format. The collected data consists of 100 medical records. Each record consists of 3 records for a test which there are 2 test for both hands, thus the total size is 1200 samples. The number of time steps for both the keyboard and sensor dataset is 500 time steps. The number of features depends on individual dataset's features. Dataset-1's input shape is (1200 × 500 × 2), Dataset-2's input shape is (1200 × 500 × 12), and Dataset-3's input shape is (1200 × 500 × 13).

B. *Neural Networks Model*

1) The simple neuron model: is created as follows.

- 3 Fully-connected neuron layers
- Output layer

The 2 hidden layer and another hidden layer has, consequently, 50 and 10 neurons dense layer with 'relu' activation function. The output layer is one fully-connected neuron with 'sigmoid' activation function. The sequential model is compiled with 'Adam' optimizer and 'binary_crossenteopy' cost function.

2) The LSTMs model: is created as follows.

- LSTMs 2 layers
- Fully-connected neuron layer
- Output layer

These two LSTMs hidden layers have initial 50 LSTMs each with 'tanh' activation function, and another layer is a 10 neurons fully-connected layer with 'linear' activation function. The output layer is a single fully-connected neuron (dense layer). The sequential model is compiled with 'SGD' optimizer and 'binary_crossentropy' owning to the binary classification problem.

In general, the output layer is set by the problem requirement. In this application, the problem is binary classification, the model output use sigmoid function as the transfer function.

There are two methods used in this work to randomly portion out the dataset into training dataset and testing dataset. The first one is known as data splitting, and the second one is cross validation.

C. *Data Splitting*

The datasets have to randomly split into training dataset and testing dataset. Each dataset is [X, y] format, the dataset is, likewise, [train X, train Y, test X, test Y] format. The test ratio is 0.2 for the simple neuron model.

K-Fold cross validation is a resampling procedure for machine learning models. A parameter, called k, is assigned as the number of groups which the dataset being separated into. This k-fold cross validation splits dataset into k groups. For each group, the k-1 datasets fit the model as training dataset, then the rest evaluates the model as testing dataset. This procedure offers the opportunity for each fold dataset to train the model for k-1 times, and to test the model for once as the hold set. In this work, the K number is 10 for the LSTMs model.

D. *Parameter and Hyperparameter Optimization*

The model parameters are variables internal the neural networks which their values are estimate from the data. Their values define the model performance. For examples, the weights and biases are the model parameters.

The model hyperparameters are variables external the neural networks which theirs values are not able to estimate directly from the data. For example, learning rate, activation function and dropout are the hyperparameters.

In this work, the Grid Search Parameter is used as the parameter optimization method. Batch Size is the number of the size for each batch. This means the total number of training data samples in a batch. The dataset is divided into smaller batches, and those batches are fed into the neural networks. The number of Iterations is the number of batches which needs to complete one epoch. The number of Epochs is the number of times for the entire dataset passing forward and backward through the neural networks once. It means the number of batches is equal to the number of iteration for one epoch.

There are many optimizers to search for the best model of neural network; for instance, 'SGD', 'RMSprop', ‘Adagrad’, ‘Adadelta’, ‘Adam’, ‘Adamax’, ‘Nadam’ and so on.

The number of Learning Rate is the number which controls the amount of updating the weight at the end of each batch. The number of Momentum is the number which controls the amount of letting the previous update influence the current weight update.

The number of Dropout Regularization is important to improve the neural networks performance. In order to avoid underfitting and overfitting model, dropout regularization impact significantly on machine learning to achieve the appropriate-fitting. The number should be between [0.0, 1.0), [1] is not possible. It generally starts at 0.2 or 20% dropout rate, and the model generates usually the robust result.

The number of Neurons in the hidden layer is one of the most important parameter to be tuned for the neural networks, because it generally represents the capacity of the neural networks.

From these optimization procedures, the final LSTMs model configuration is now as follows:

- 3 hidden layers :
  - 100 LSTMs with ‘relu’ activation function, ‘uniform’ weight initializer, maxnorm(4) weight constraint and 0.2 dropout rate for 2 layers
  - 50 neurons with ‘sigmoid’ activation function and ‘uniform’ weight initializer dense layer
- Output layer : 1 neuron dense layer
- Compiling the model with ‘binary_crossentropy’ loss function and ‘Adam’ optimizer.
- Batch size: 100, 200 epochs, 0.01 learning rate.

E. *Evaluating the Model*

There are two methods of evaluating the models which are Hold-Out and Cross-Validation for large dataset and limited amount of dataset consequently. The hold-out method is randomly divided data into three subsets: Training set, Validation set and Testing set.

In this application, K-fold cross-validation for the LSTMs model due to amount of data availability is used to compare with traditional data splitting with 0.2 test ratio for the simple neuron model.

III. RESULTS

There are two summarized result from initial setting and optimized setting. The records of accuracy are shown in percentage. Each result is obtained from the exported model with the Parkinson’s disease data.

The first result, the simple neuron model is trained with simple training and testing dataset with 0.2 ratio test data splitting and initial configuration as shown in Table I.

TABLE I. THE RESULTS FROM THE SIMPLE FULLY-CONNECTED NEURON NEURAL NETWORK WITH NORMAL DATA SPLITTING AND DEFAULT PARAMETER SETTING.

Results	Dataset1	Dataset2	Dataset3
1	53.4198	57.192	62.541
2	54.1945	60.0691	63.1512
3	52.9121	61.894	64.0891
4	56.1148	62.1905	61.261
5	54.2189	59.1892	59.198
6	54.2105	58.1984	59.0148
7	58.641	60.1444	60.4717
8	55.0001	58.176	61.2239
9	56.1894	59.6697	62.3617
10	53.4234	59.9963	63.7879

The second result, the LSTMs model is trained with K-fold cross-validation dataset with optimum parameter configuration from parameter optimization procedure as shown in Table II.

TABLE II. THE RESULTS FROM THE LSTMS NEURAL NETWORK WITH K-FOLD CROSS-VALIDATION AND OPTIMIZED PARAMETER SETTING.

Results	Dataset1	Dataset2	Dataset3
1	58.7846	67.8156	70.5491
2	61.3605	68.1602	72.911
3	62.4908	72.4869	73.6619
4	57.096	74.3201	75.6138
5	59.5541	70.7767	74.9995
6	63.0847	71.591	73.1542
7	60.3337	73.4852	75.6173
8	58.6913	69.5894	76.2189
9	62.7185	72.5507	74.0695
10	59.1515	73.0564	72.0519

The results from both configurations are distinct. The performance of the LSTMs neural network with k-fold cross validation and parameter optimization is clearly an improvement over the simple Neural Network. Using dataset 3 is the best dataset 3 for this binary classification. The comparing between simple neuron model and LSTMs model is shown in Table III.

TABLE III. EXPERIMENT RESULT FOR DATASET3

	Simple Neuron (%)	LSTMs (%)
Mean	61.7100	73.8847
Max. Acc.	64.0891	76.2189
Min. Acc.	59.0148	70.5491
Std.	1.7862	1.7868

The machine learning using k-fold cross-validation and parameter optimization can achieve the best result at 76.22% with 73.88% average and 70.55% minimum. The result is better than the NN configuration for at least 10% in average. This comparison between the simple neuron and LSTMs model represent the advantage of using LSTM for time series data.

IV. DISCUSSION

There are several limiting factors in this dataset which prevent machine learning performance.

- Parkinson's disease patients have one dominant hand showing distinct symptom and another hand being like if normal person.
- Symptoms of some Parkinson's disease patients are almost healthy as normal, they only have slightly sickness which indicate to be diagnosed as Parkinson's disease.
- Amount of data is not adequate to train the deep neural networks so as to receive higher accuracy classification prediction.

V. CONCLUSION

The LSTM neural network with K-Fold cross validation is an appropriate approach for time-series binary classification. This machine learning method works not only for Parkinson's Disease data but also other time series data classification problems. The LSTM model can be improved to achieve higher accuracy using improved feature extraction. The experiment shows promising results that the proposed method can be used for early screening of patients. This method needs further improvement before it can be used by OPD doctors for Parkinson's disease diagnosis.

REFERENCES

[1] Akin Ozcift, and Arif Gulten, "Classifier ensemble construction with rotation foreast to improve medical diagnosis performance of machine learning algorithms," Computer Methods and Programs in Biomedicine 104, 2011, pp.443-451.

[2] P. R. Davidson, R. D. Jones1,and M. T. R. Peiris, "Detecting Behavioral Microsleeps using EEG and LSTM Recurrent Neural Networks," the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, 2005.

[3] Zachary C. Lipton, David C. Kale, Charles Elkan, and Randall Wetzel, "Learning to diagnose with LSTM recurrent neural networks," arXiv:1511.03677v7, 2017.

[4] Najmeh Nazari, Seyed Ahmad Mirsalari, Sima Sinaei, Mostafa E. Salehi, and Masoud Daneshtalab, "Multi-level Binarized LSTM in EEG Classification for Wearable Devices," 28th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, 2020.

[5] Inci M. Baytas, Cao Xiao, Xi Zhang, Fei Wang, Anil K. Jian, and Jiayu Zhou, "Patient Subtyping via Time-Aware LSTM Networks," KDD, 2017.

[6] Daniel Soutner , and Ludek Muller, "Application of LSTM Neural Networks in Lanfguage Modelling," Springer-Verlag Berlin Heidelberg, 2013, pp.105-112.

[7] John Gamboa, "Deep Learing for time-series analysis," arXiv:1701.01887v1, 2017.

[8] Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward, "Deep Sentence Embedding Using Long Short-Term Memory Networks: Analysis and Application to Information Retrieval," IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol.24, No.4, 2016.

[9] Hasim Sak, Andrew Senior, and Francoise Beaufays, "Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling," Interspeech, 2014.

[10] Xiaopeng Xi, Eamonn Keogh, Christian Shelton, and Li Wei, "Fast Time Series Classification Using Numerosity Reduction," the 23rd International Conference on Machine Learning, 2006.

[11] Wonmin Byeon, Thomas M. Breuel, Federico Raue, and Marcus Liwicki, "Scene Labeling with LSTM Recurrent Neural Networks," IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[12] Fazle Karim, Somshubra Majudar, Houshang Darabi and Shen Chen, "LSTM Fully Convolutional Networks for Time Series Classification," IEEE Access, Vol.6, 2017.

[13] Shivam Chandhok, Himani Joshi, S. J. Darak, and A. V. Subramanyam, "LSTM Guided Modulation Classification and Experimental Validation for Sub-Nyquist Rate Wideband Spectrum Sensing," 11th International Conference on Communication Systems and Networks, 2019.

[14] Pierre Geurts, "Pattern Extraction for Time Series Classification," PKDD, 2001, pp.115-127.

[15] Hui-Ling Chen, Chang-Cheng Huang, Xin-Gang Yu, Xin Xu, Xin Sun, Gang Wang, and Su-Jing Wang, "An efficient diagnosis system for detection of Parkinson's disease using fuzzy k-nearest neighbor approach," Expert Systems with Application 40, 2013, pp.267-271.

[16] Hui-Ling Chen, Gang Wang, Chao Ma, Zhen-Nao Cai, and Wei-Bib Liu, "An efficient hybrid kernel extream learning machine approach for early diagnosis of Parkinson's disease," Neurocomputing 184, 2016, pp.131-144.

[17] Chao Che, Cao Xiao, Jian Liang, Bo Jin, Jiayu Zhou, and Fei Wang, "An RNN Architecture with Dynamic Temporal Matching for Personalized Predictions of Parkinson's Disease," SIAM International Conference on Data Mining, 2017.

[18] Ram Deepak Gottapu, and Cihan H. Dagli, "Analysis of Parkinson's Disease Data," Procedia Computer Science 140, 2018, pp.334-341.

[19] Athanasios Tagaris, Dimitrios Kollias, and Andreas Stafylopotis, "Assesment of Parkinson's Disease Based on Deep Neural Networks," Springer International Publicshing AG, 2017, pp.391-403.

[20] Xi Zhang, Jingyuan Chou, Jian Liang, Cao Xiao, Yize Zhao, Harini Sarva, Claire Henchcliffe, and Fei Wang, "Data-driven subtyping of Parkinson's disease using longtitudinal clinincal cecords: A Cohort study," Scientific Reports, 2019.

[21] David Gil A., and Magnus Johnson B., "Diagnosing Parkinson by using Artificial Neural Networks and Support Vector Machines," Global Journal of Computer Science and Technology 9, 2009.

[22] Shreya Bhat, U. Rajendra Acharya, Yuki Hagiwara, Nahid Dadmehr, and Hojjat Adeli, "Parkinson's disease: Cause factors, measurable indicators, and early diagnosis," Computers in Biology and Medicine 102, 2018, pp.234–241.

[23] Siddharth Arora, Fahd Baig, Christine Lo, Thomas R. Barber, Michael A. Lawton, Andong Zhan, Michal Rolinski, Claudio Ruffmann, Johannes C. Klein, D, Jane Rumbold, BSc, Amandine Louvel, Zenobia Zaiwalla, Graham Lennox, Tim Quinnell, Gary Dennis, Richard Wade-Martins, Yoav Ben-Shlomo, Max A. Little, and Michele T. Hu, "Smartphone motor testing to distinguish idiopathic REM sleep behavior disorder, controls, and PD," Neurology, 2018.

[24] dong Zhan, Srihari Mohan, Christopher Tarolli, Ruth B. Schneider, Jamie L. Adams, Saloni Sharma, Molly J. Elson, Kelsey L. Spear, Alistair M. Glidden, Max A. Little, Andreas Terzis, E. Ray Dorsey, Suchi Saria, "Using Smartphones and Machine Learning to Quantify Parkinson Disease Severity The Mobile Parkinson Disease Score," JAMA Neurology, 2018.

[25] Resul Das, "A comparison of multiple classification methods for diagnosis of Parkinson disease," Expert Systems with Application 37, 2010, pp.1568-1572.