

รายงานสรุปหัวข้อการค้นคว้า

Forex exchange rate forecasting
using deep recurrent neural networks

จัดทำโดย

นายปณณวิชญ์ พันธวงศ์ 600610752

เสนอ

รศ.ดร.คันสนีย์ เอื้อพันธ์วิริยะกุล

รายงานเล่มนี้เป็นส่วนหนึ่งของวิชา

261456

ภาคเรียนที่ 1 ปีการศึกษา 2563

สารบัญ

สารบัญ	1
OVERVIEW.....	3
1. NEURAL NETWORK ARCHITECTURES	4
1.1 Neural network architectures	4
1.1.1. Feedforward Neural Networks (FNN)	4
1.1.2. Recurrent Neural Networks (RNN)	4
1.1.3. Long Short-Term Memory networks (LSTM)	5
1.1.4. Gated Recurrent Units (GRU)	6
2. EXPERIMENTAL DESIGN	7
2.1 Data	7
2.2 Data preprocessing.....	7
2.2.1 Length of the input sequences Training and trading window	7
2.2.2 Feature	8
2.2.3 Targets	8
2.2.4 Loss function	8
2.2.5 Activation function	9
2.2.6 Regularization	9
2.2.7 Regularization	9
3. MODEL ARCHITECTURE.....	11
3.1 Train data	11

3.2 Test data	12
3.3 Architecture	12
4. EVALUATION	13
4.1 การวัดความแม่นยำของการทำนาย	13
4.1.1 Logarithmic loss (Log loss)	13
4.1.2 Predictive accuracy (Acc.)	13
4.1.3 Area under the receiver operator characteristic curve (AUC)	13
4.2 การวัดความเสี่ยงในการลงทุน	13
4.2.1 Sharpe ratio (SR)	13
4.2.2 Standard deviation (SD)	13
4.3 การวัดผลตอบแทนสุทธิ	13
4.4 ผลการทดลอง	14
5. CONCLUSION	16
REFERENCE	17

Overview

ในงานวิจัยนี้ได้นำเสนอเกี่ยวกับการใช้ Artificial Neural Network (ANN) ซึ่งประกอบไปด้วย Feedforward neural Networks (FNN), Recurrent Neural Networks (RNN), Long Short-Term Memory Networks (LSTM) และ Gated Recurrent Units (GRU) ในการเปรียบเทียบการทำนาย Classification ราคาของตลาดฟอเร็กซ์ ใน 4 คู่สกุลเงิน ประกอบไปด้วย EUR/USD ,GBP/USD ,USD/JPY และ USD/CHF โดยได้สรุปผลในรูปแบบ 2 มาตรวัดผลของแต่ละ Artificial Neural Network (ANN) ได้แก่ มาตรวัดผลความแม่นยำของผลทำนาย และ มาตรวัดผลในการลงทุน ซึ่งงานวิจัยนี้ยังได้เสนอแนวทางในการพัฒนาและปัญหาที่พบในขณะทำการทดลองเพื่อให้งานวิจัยในอนาคตสามารถนำไปพัฒนาต่อได้อีกด้วย

1. Neural network architectures

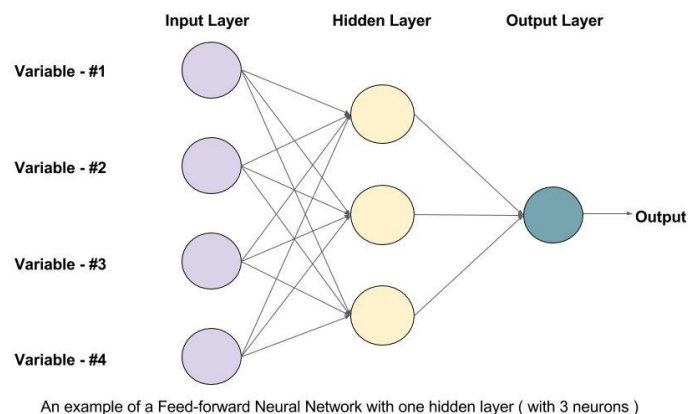
ในหัวข้อนี้จะกล่าวถึงโครงสร้างของ Neural network ที่ใช้สำหรับงานวิจัยนี้ซึ่งประกอบไปด้วย Feedforward neural Networks (FNN), Recurrent Neural Networks (RNN), Long Short-Term Memory Networks (LSTM) และ Gated Recurrent Units (GRU)

1.1 Neural network architectures

ในงานวิจัยนี้จะใช้โครงสร้างของ Artificial Neural Network ในการทดลองทั้งหมด 4 โครงสร้าง

1.1.1. Feedforward Neural Networks (FNN)

คือโครงข่ายประสาทเทียมที่เป็นการจำลองมาจากสมองของเรา แต่ในส่วนคอมพิวเตอร์นั้นไม่ได้มีโครงข่ายที่ซับซ้อนเหมือนกับสมองของเรา มันมีหน้าที่แค่รันโปรแกรมตามคำสั่งของเราเท่านั้น ดังนั้นเมื่อเราจะให้มันทำการเรียนรู้อะไรสักอย่างจึงเป็นเรื่องยากในรูปแบบปกติ จึงเกิดการจำลองแนวทางการเรียนรู้ของคนไปสู่คอมพิวเตอร์ด้วย Neural Network นั่นเอง



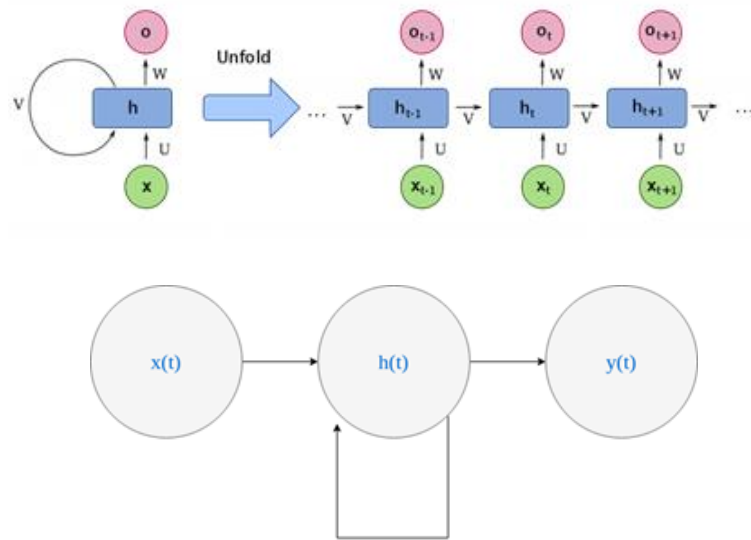
รูปที่ 1 แสดงตัวอย่างโครงสร้างของ FNN

(<https://www.learnopencv.com/>)

1.1.2. Recurrent Neural Networks (RNN)

คือ Artificial Neural Network แบบหนึ่งที่ย่อแบบมาแก้ปัญหาสำหรับงานที่ข้อมูลมีลำดับ Sequence ชื่อ Recurrent มันก็แปลตรงตัวแล้วว่าจะมี “การเกิดซ้ำ” กล่าวคือ มีการเรียก hidden state จาก state ก่อนๆมา เช่น เรียกข้อมูลจาก Hidden state $h(t-1)$ ที่ได้จาก

การคำนวณจากข้อมูล $x(t-1)$ มารวมคำนวณกับ Hidden state $h(t)$ ที่ได้จากการคำนวณจากข้อมูล $x(t)$ ด้วยนั่นเอง

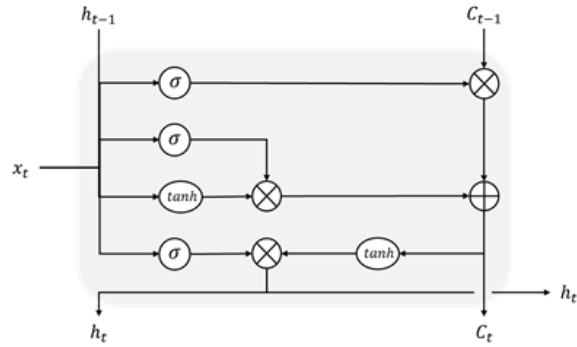


รูปที่ 2 แสดงโครงสร้างการทำงานของ RNN

(<https://www.bualabs.com/>)

1.1.3. Long Short-Term Memory networks (LSTM)

LSTM เป็น Neural Network ถูกพัฒนามาจาก RNNs หลักการของ LSTMS คือการเก็บ 'สถานะ' ของแต่ละ node เอาไว้ด้วย เพื่อว่าตอนย้อนกลับมาจะได้รู้ค่านี้แท้จริงแล้วเป็นอะไรมาก่อน สิ่งที่ทำให้ LSTMs โดดเด่นขึ้นมานั้นก็คือการที่มันสามารถเลือกได้ว่า ข้อมูลไหนที่ควรจะต้องจำ ข้อมูลไหนที่ควรจะต้องกำจัดทิ้งออกไป ผ่านการ 'สลิ้ม' ของสถานะใน node นั้น ๆ Forget gate คือ ประตูที่จะใช้เป็นตัวกำหนดว่า ข้อมูลที่เข้ามานั้นสมควรที่จะได้ออกไปทั้งหมดหรือไม่.

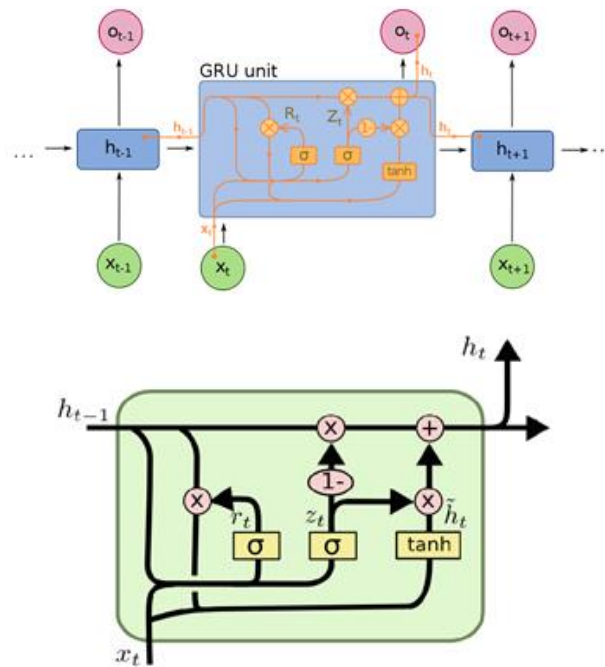


รูปที่ 3 แสดงโครงสร้าง Forget gate

(<https://medium.com/>)

1.1.4. Gated Recurrent Units (GRU)

อัปเดตมาจาก RNNs เป็นกลไกเปิดปิดการอัปเดตสถานะภายใน Recurrent Neural Network ที่คล้ายกับ Long Short-Term Memory (LSTM) ที่จะมี Forget Gate แต่มี Parameter น้อยกว่า LSTM เนื่องจากไม่มี Output Gate



รูปที่ 4 แสดงโครงสร้างของ GRU

(<https://medium.com/>)

2. Experimental design

ในหัวข้อนี้จะกล่าวถึงกระบวนการทดลองของงานวิจัยนี้ซึ่งประกอบไปด้วย คำอธิบายข้อมูล, วิธีการเตรียมข้อมูล รวมถึงการกำหนดค่าไฮเปอร์พารามิเตอร์

2.1 Data

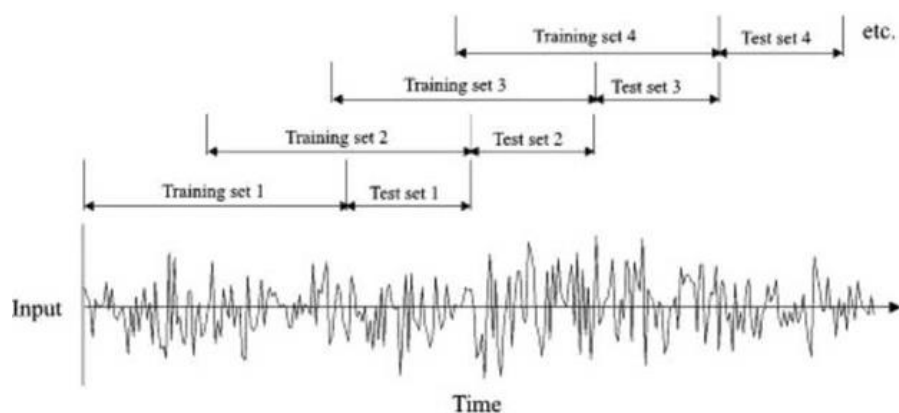
เป็นชุดข้อมูลของ 4 คู่สกุลเงินหลัก ได้แก่ EUR/USD ,GBP/USD ,USD/JPY และ USD/CHF ข้อมูลของแต่ละคู่สกุลเงินมี 12,710 แถว ที่เป็นข้อมูลกรอบเวลา 1 วัน อยู่ในช่วงเวลา 4 ม.ค. 1971 จนถึง 25 ส.ค. 2017 ที่ใช้งานวิจัยก่อนหน้านี้ (Kuan and Liu 1995; Tenti 1996; Giles et al. 2001; Kiani and Kastens 2008; Hussain et al. 2008)

2.2 Data preprocessing

2.2.1 Length of the input sequences Training and trading window

ทำการแบ่งข้อมูลสำหรับการเทรน และสำหรับทดสอบ โดย sequences length สำหรับอินพุตมีความยาว $T = 240$ และทำการแบ่งข้อมูลเป็นทั้งหมด 4 ชุด (van de Gucht et al. 1996)

โดยชุดที่ 1 จะนำข้อมูลบางส่วนจากชุดที่ 2 มาใช้เทรนและทดสอบ , ชุดที่ 2 จะนำข้อมูลบางส่วนในชุดที่ 3 มาใช้เทรน และทดสอบ ชุดต่อไปก็จะทำแบบเดียวกันกับชุด 1 และ 2 จนไปถึงชุดที่ 4 ซึ่งกระบวนการนี้เรียกว่า the window-based cross-validation



รูปที่ 5 แสดงการแบ่งข้อมูล

2.2.2 Feature

ใช้เพียงแค่ 1 feature เป็นอัตราแลกเปลี่ยนที่แสดงถึงราคาเมื่อเปรียบเทียบกับสกุลเงิน USD โดยคำนวณจากร้อยละการเปลี่ยนแปลงของราคาจากวันที่ t

$$r_t^c = \frac{P_t^c}{P_{t-1}^c} - 1$$

โดยให้ P_t^c คือ ราคาของสกุลเงิน c เมื่อเปรียบเทียบกับสกุลเงิน USD
 r_t^c คือ ร้อยละการเปลี่ยนแปลงของราคาจากวันที่ t

โดยอินพุตที่จะเข้าเทรนในโมเดล จะประกอบไปด้วย ลำดับของร้อยละการเปลี่ยนแปลงของราคาจากเวลาที่ t ของตั้งแต่วันที่ จนวันที่ t

$$X_t^c = \{r_{1-t}^c, r_{1-t+1}^c, r_{1-t+2}^c, \dots, r_{t-1}^c\}$$

2.2.3 Targets

เป็นปัญหา binary classification problem ซึ่ง target จะเป็นร้อยละการเปลี่ยนแปลงของราคาวันถัดไป โดยอ้างอิงงานวิจัย Fischer and Krauss (2018) จะกำหนดเป็นดังนี้

$$Y_t^c = \begin{cases} 1 & \text{if } r_t^c \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

โดยถ้าค่า r ถ้ามากกว่าหรือเท่ากับ 0 จะกำหนดให้เป็น 1 และกรณีอื่นๆให้เป็น 0

2.2.4 Loss function

ซึ่งในการเทรนจะใช้ cross-entropy ระหว่าง ค่าที่ทำนายและค่าจริง

$$L_S(y_{T_S}, \hat{y}_{T_S}) = -\frac{1}{|T_S|} \sum_{t \in T_S} (y_t \log(\hat{y}_t) + (1 - y_t) \log(1 - \hat{y}_t))$$

2.2.5 Activation function

ซึ่งแต่ละนิวรอนเน็ตเวิร์คใช้ activation function ดังนี้

โครงสร้าง	Activation ภายใน	Activation output
FNN	ReLu	Sigmoid
RNN	Tanh	
LSTM	Sigmoid,Tanh	
GRU		

โดย FNN ใช้ Activation function เป็น ReLu เนื่องจากต้องการปรับปรุงการไหลของ Gradient และ ลดระยะเวลาในการเทรน

ในเอาต์พุตของแต่ละโครงสร้างนั้นใช้ Activation function เป็น Sigmoid เนื่องจากค่าอยู่ในช่วง $[0, 1]$ ซึ่งเป็นค่าของความน่าจะเป็น

2.2.6 Regularization

ใช้วิธีการ Dropout เป็นการสุ่มถอดบาง neurons ออก ในระหว่างการเทรน

2.2.7 Regularization

วิจัยนี้ได้ทำการทดลองเปลี่ยนค่าตามตารางดังกล่าว และได้สรุปผลในการหาโครงสร้างที่ดีที่สุดของแต่ละโดเมนในหัวข้อถัดไป

ชื่อ	ปริมาณ
Hidden Layer	1,2,3,4
Neurons per Hidden Layer	25,50,100,200,400,800,1600

Dropout	0 ถึง 60% , เพิ่มทีละ 10%
Optimizer and learning rate	Adam และ RMSprop
Batch size	16,32,64,128,256

3. Model architecture

3.1 Train data

โดยรายละเอียดของการเทรนของแต่ละโครงสร้างนั้นมีรายละเอียดดังนี้

รายการ	รายละเอียด
ปริมาณข้อมูล	750
ความยาวอินพุต	240 (ข้อ 2.2.1) *สำหรับ FNN ความยาว คือ 1
ระยะข้อมูล	-1 ถึง 1
Mini batch size	32
Optimizer	Adam(default parameters)
Epochs	100 epochs โดยหยุดก่อน กำหนดหลังจาก 10 ช่วงเวลาโดย ไม่ต้องปรับปรุง validation loss 20%

3.2 Test data

ใช้ข้อมูลสำหรับทดสอบ 250 ข้อมูล โดยทำการ Preprocessing ตามข้อ 2.2.3

3.3 Architecture

สำหรับทุกๆโครงสร้างมีรายละเอียดดังนี้

Hidden Layer	3
Neurons Per Hidden Layer	50
Dropout	25% หลังแต่ละ Hidden Layer

4. Evaluation

สำหรับการวัดผลของงานวิจัยนี้ ได้ทำการวัดผลในทั้งหมด 2 มาตรวัดได้แก่ การวัดความแม่นยำของระบบ และ การวัดผลในการลงทุน

4.1 การวัดความแม่นยำของการทำนาย

ใช้วิธีการวัดความแม่นยำทั้งหมด 3 รูปแบบ ดังนี้

4.1.1 Logarithmic loss (Log loss)

เป็น loss function สำหรับ minimize ขณะกำลังเทรน

4.1.2 Predictive accuracy (Acc.)

ใช้จำแนกประสิทธิภาพในการแบ่ง class

4.1.3 Area under the receiver operator characteristic curve (AUC)

ซึ่งถ้าหากพื้นที่ใต้โค้ง ROC มาก จะหมายถึงมีความถูกต้องมากตาม

4.2 การวัดความเสี่ยงในการลงทุน

4.2.1 Sharpe ratio (SR)

ใช้สำหรับเป็นมาตรวัดความเสี่ยงในการลงทุน

4.2.2 Standard deviation (SD)

ค่าเบี่ยงเบนมาตรฐานใช้วิเคราะห์ผลตอบแทนการลงทุน

4.3 การวัดผลตอบแทนสุทธิ

ในงานวิจัยนี้จะกล่าวถึง ผลตอบแทนที่ได้(R_s) (ระยะเวลาประมาณ 1 ปี) จากชุดทดสอบประกอบไปด้วยข้อมูล 240 วันซื้อขาย โดยใช้สมการดังนี้

$$R_s = \prod_{t \in T_s} (1 + \tilde{r}_t^c) - 1$$

4.4 ผลการทดลอง

Table 3 Results from a naive forecast by time series and aggregated (average weighted by length of time series)

	Acc.	AUC	Returns	SD	SR
EUR/USD	0.4744	0.4718	-0.0202	0.0060	-0.0188
GBP/USD	0.5010	0.4971	0.0481	0.0059	0.0310
USD/JPY	0.4940	0.4888	0.0488	0.0063	0.0280
USD/CHF	0.4873	0.4839	0.0131	0.0071	0.0014
Weighted Avg.	0.4921	0.4880	0.0307	0.0064	0.0161

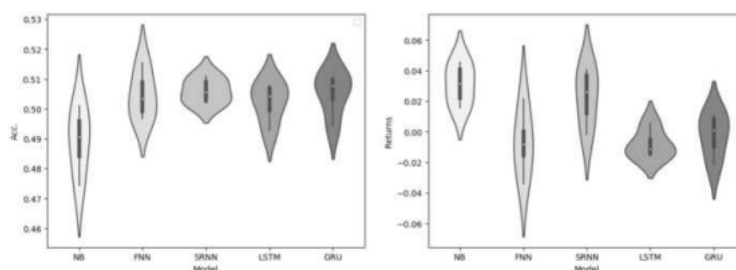


Fig. 11 Accuracy and trading strategy returns of the naive benchmark and the four deep learning models

จากตารางที่ 3 เป็นผลลัพธ์การทำนายจากงานวิจัย พบว่า คู่สกุลเงิน GBP/USD มีความแม่นยำเป็นอันดับหนึ่งเมื่อเปรียบเทียบกับคู่สกุลเงินอื่นๆ และอันดับสุดท้ายคือ คู่สกุลเงิน EUR/USD ที่มีความแม่นยำต่ำสุด

Table 4 Results for the four neural networks by currency pair and model type averaged across the test windows in our window-based cross-validation approach

Time Series	Model	Log Loss	Acc.	AUC	Returns	SD	SR
EUR/USD	FNN	0.6953	0.5155	0.5202	0.0218	0.0060	0.0186
	SRNN	0.7114	0.5019	0.5003	0.0406	0.0060	0.0240
	LSTM	0.6948	0.4928	0.5005	-0.0138	0.0060	-0.0073
	GRU	0.6948	0.4944	0.5103	-0.0216	0.0060	-0.0131
GBP/USD	FNN	0.6964	0.5068	0.5035	-0.0094	0.0059	-0.0034
	SRNN	0.7064	0.5110	0.5116	0.0166	0.0059	0.0098
	LSTM	0.6943	0.5066	0.5021	-0.0088	0.0059	-0.0041
	GRU	0.6945	0.5064	0.4930	-0.0056	0.0059	-0.0021
USD/JPY	FNN	0.7001	0.4966	0.4995	-0.0340	0.0063	-0.0255
	SRNN	0.7100	0.5030	0.4955	-0.0019	0.0063	-0.0081
	LSTM	0.6956	0.5019	0.5077	-0.0157	0.0063	-0.0143
	GRU	0.6945	0.5091	0.5089	0.0075	0.0038	0.0092
USD/CHF	FNN	0.6977	0.4999	0.4982	-0.0068	0.0071	-0.0019
	SRNN	0.7016	0.5081	0.5057	0.0356	0.0071	0.0196
	LSTM	0.6936	0.5079	0.5080	0.0056	0.0071	0.0044
	GRU	0.6941	0.5108	0.5109	0.0108	0.0071	0.0057
Weighted Avg.	FNN	0.7026	0.5062	0.5061	-0.0126	0.0064	-0.0071
	SRNN	0.7115	0.5103	0.5073	0.0195	0.0064	0.0090
	LSTM	0.6993	0.5076	0.5088	-0.0072	0.0064	-0.0050
	GRU	0.6992	0.5107	0.5085	0.0014	0.0057	0.0024

ตารางที่ 4 แสดงค่าเฉลี่ยของการทดสอบในรูปแบบ multiple test windows ดังที่กล่าวในหัวข้อ 2.2.1 Length of the input sequences, Training and trading window ซึ่งสรุปได้ 3 ประการดังนี้

- **Training loss (Log Loss)**

ในส่วนของ LSTM และ GRU ทำงานได้ดีกว่า FNN และ RNN ในแต่ละคู่สกุลเงิน เพียงเล็กน้อย เนื่องจาก LSTM และ GRU เหมาะสมกับข้อมูลที่มีลำดับ sequences ยาวๆ

- **Economic measures**

พบว่าไม่มีโมเดลไหนเลยที่สามารถให้ผลตอบแทนในทางที่ดีได้ ทั้งในแง่ของผลตอบแทนสุทธิ และความเสี่ยงในการลงทุน ซึ่งจากตารางที่ 4 จะเห็นได้ว่า RNN มีประสิทธิภาพเทียบเท่ากับ LSTM และ GRU

- **AUC**

ในส่วนของพื้นที่ใต้รูปโค้งพบว่า มีค่าต่างกันไม่มากนัก ซึ่งในแต่ละคู่สกุลเงิน ก็จะมีโมเดลที่ทำได้ดีที่แตกต่างกันไป

5. Conclusion

งานวิจัยนี้ทำการทดลอง Forex classification โดย Target ที่กำหนดไว้ คือ ร้อยละการเปลี่ยนแปลงของราคาวันถัดไป ซึ่งเป็นประเภท Binary classification ซึ่งมีการทดลองทำทั้งหมด 4 โมเดล ได้แก่ FNN,RNN,LSTM และ GRU โดยมี feature คือ ร้อยละการเปลี่ยนแปลงของราคาในเวลา t ก่อนหน้า ซึ่งมาตรวจวัดผล หรือ การวัดความแม่นยำของระบบที่จะ 2 กลุ่มหลักๆ คือ มาตรวจวัดความแม่นยำของระบบ และมาตรวจวัดในทางการเงิน โดยผลสรุปที่ได้ในการทดลองพบว่า ทุกๆโมเดลมีความแม่นยำในการทำนาย(Acc) อยู่ที่ประมาณ 50% และมีความเสี่ยงในการลงทุนค่อนข้างสูง ซึ่งขอเสนอแนะสำหรับงานวิจัยในอนาคตมีดังนี้

- **Feature**

feature ที่ใช้มีเพียงแค่ค่าของร้อยละการเปลี่ยนแปลงของราคา ซึ่งอาจจะเป็น feature ที่น้อยไปหรือไม่เหมาะสมกับการทำนายตลาดฟอเร็กซ์

- **Forex market**

เนื่องจากตลาด Forex มีความผันผวนค่อนข้างสูงกว่าตลาดการเงินอื่นๆ ทำให้ความแม่นยำของระบบทำนายไม่สามารถคาดคะเนได้

- **Amounts of data**

ปริมาณข้อมูลอาจจะมีผลต่อการเรียนรู้ของระบบซึ่งหากน้อยเกินไปก็จะทำให้ผลของการทำนายไม่ดีนัก

Reference

- [1] Kuan, C. M., & Liu, T. (1995). Forecasting exchange rates using feedforward and recurrent neural networks. *Journal of Applied Econometrics*, 10(4), 347–364
- [2] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1), 1929–1958.
- [3] Giles, C. L., Lawrence, S., & Tsoi, A. C. (2001). Noisy time series prediction using recurrent neural networks and grammatical inference. *Machine Learning*, 44(1), 161–183.
- [4] Kiani, K. M., & Kastens, T. L. (2008). Testing forecast accuracy of foreign exchange rates: Predictions from feed forward and various recurrent neural network architectures. *Computational Economics*, 32(4), 383–406.
- [5] Hussain, A. J., Knowles, A., Lisboa, P. J. G., & El-Deredy, W. (2008). Financial time series prediction using polynomial pipelined neural networks. *Expert Systems with Applications*, 35(3), 1186–1199
- [6] van de Gucht, L. M., Dekimpe, M. G., & Kwok, C. C. Y. (1996). Persistence in foreign exchange rates. *Journal of International Money and Finance*, 15(2), 191–220.
- [7] Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2), 654–669.
- [8] Alexander Jakob Dautel, Wolfgang Karl Härdle, Stefan Lessmann, Hsin-Vonn Seow . (2020). Forex exchange rate forecasting using deep recurrent neural networks.