Annexure 1: Implementation Results of the Systematic Enhanced Deep Learning Framework for Irregular Sequential Analysis (SeLFISA)

Section A: Results of Irregular Sequential Patterns Analysis

List	Name	Length	Number of IQR Outliers
1	AUD_USD DailyExchaRate from 1990-2016 by Chang et al.(2018)	5906	0
2	GBP_USD DailyExchangeRate from 1990-2016 by Chang et al.(2018)	6135	639
3	CAD_USD DailyExchangeRate from 1990-2016 by Chang et al.(2018)	5000	0
4	SwiFranc_USD DailyExchangeRate from 1990-2016 by Chang et al.(2018)	7015	0
5	CNY-USD DailyExchangeRate from 1990-2016 by Chang et al.(2018)	5000	0
6	NZD_USD DailyExchangeRate from 1990-2016 by Chang et al.(2018)	5000	0
7	JPY_USD DailyExchangeRate from 1990-2016 by Chang et al.(2018)	5000	24
8	SGD_USD DailyExchangeRate from 1990-2016 by Chang et al.(2018)	5000	0
9	Monero CryptoCurrencyDailyRates from 2015-2018 by Glenski et al. (2019)	1208	0
10	S&P from 01-2008_12-2009 by Chalvatzisa et al.(2019)	504	0
11	DJI from 01-2008_12-2009 by Chalvatzisa et al.(2019)	504	0
12	NASDAQ from 01-2008_12-2009 by Chalvatzisa et al.(2019)	504	0
13	S&P500 from 10-2010_09-2016 by Bao et al. (2017) and Chalvatzisa et al.(2019)	1696	0
14	DJI from 10-2010_09-2016 by Bao et al. (2017) and Chalvatzisa et al.(2019)	1513	0
15	NASDAQ from Jan-Dec_2011 by Zhou et al. (2019) and Chalvatzisa et al.(2019)	251	6
16	S&P500 from Jan-Dec_2011 by Zhou et al. (2019) and Chalvatzisa et al.(2019)	251	0

Section B: Results of Variables for SeLFISA Framework

*This is based on the systematic literature review outcomes and experimental work guided by SeLFISA Framework

List	Variable Name	Internal Features	Description of Internal Features	Category	Remarks
1.	Domain of research	1	Deep Learning Framework for the Analysis of Discrete Irregular Patterned Sequential Environments	Combinatorial in identification and selection but permutative in implementation	This is the initial stage driven by the research challenges in sequential modelling.
2.	Domain Challenges classes	3	Major classes are within frameworks, datasets and evaluation	Combinatorial	Addressed all through a framework
3.	Specific analysis challenges	11	consistency or inconsistency, reliability, repeatability, straightforwardness transparency, explainability, sensitivity to outliers and extreme values, lack of well-established, explainable literature, poor comprehensive comparison analysis, lack of multidimensional performance evaluation on single framework, dominance of accuracy metrics, computational complexity.	Combinatorial	Focused on those that distort performance robustness
4.	Existing research sources	400	400 articles were the initial sources of literature research	Combinatorial	33 articles created nucleus articles based on a matrix specific selection, inclusion and analysis criteria.
5.	Implementation AI Platforms	8	Google AI Cloud Platform, Amazon AI Services (Amazon SageMaker), Google Cloud AutoML, MATLAB, Microsoft Azure (Machine Learning Studio), IBM Watson Machine Learning and Anaconda Enterprise.	Combinatorial	Anaconda Enterprise was our platform of choice because its open source versatility platform with a Python based IDE compatibility with many languages and notebooks. This avail the entire life cycle which prepare, build, validate, deploy and monitor AI models.
6.	Implementation	5	Python, Java, Lisp, Prolog and R Programming	Combinatorial	Python was our language of choice since it is easy to

List	Variable Name	Internal Features	Description of Internal Features	Category	Remarks
	Languages				learn, deploy and it integrates efficiently with a wide range of syntaxes
7.	Implementation environments	3	Jupyter Notebook, Kaggle and Google Colaboratory	Combinatorial	We created our environment based on Jupyter Notebook because of its ability to XX
8.	Libraries and modules	22	Regular expression, garbage collectors, operating system, system-specific parameters, time, spacy, Keras, pickle, requests, math time, Matplotlib, NumPy, Pandas, progress bar TQDM library, math log2, Seaborn, sklearn, metrics, TensorFlow.	Combinatorial	We choose more than 22 libraries and modules that are already written in Python to set routines and functions. These libraries and modules were expanded from internal module through an "**from** main library **import** internal library'
9.	Computational Environment	4	High Performance Computing from CHPC, Google Cloud, Kaggle and On-Premise Core i7 Laptop.	Combinatorial on design and Permutative on installation and executions	CHPC High Performance Computing combined through on-Prem Laptop.
10.	Datasets domain	8	Weather, energy, finance, weather, astronomy, transportation, health and general domain benchmark datasets.	Combinatorial	Finance domain was our primary choice.
11.	Datasets	73	The 8 domains from 33 nucleus articles produced 73 accessible datasets.	Combinatorial	2 Financial market-daily currency exchange datasets were selected with high levels of irregular discrete properties.
12.	Selected dataset features	6	Date, price, open, high, low and change	Combinatorial	Preprocess before training.
13.	Data exploratory processes	10	More than 10 activities in the form of data wrangling, description, data pre-processing, data munching, data cleaning, and exploratory data analysis	Permutative	Preprocess before training.
14.	Dataset splitting ratio	3	Training, validation and testing (80%-20%, 90%-10% and 70%-30%)	Combinatorial on ratio selection and permutative on execution	Preprocess before training.
15.	Algorithms and models	335	These architectures produced 335 algorithms and models based on	Combinatorial on selection and	Focused on best performing through experimental deployment and application

List	Variable Name	Internal Features	Description of Internal Features	Category	Remarks
			statistical, probabilistic, gated, attention, bidirectional, general neural, encoders and decoders, transformer, vanilla, hybrid, ensemble, convolutional, classification and other	Permutative during execution	
16.	Algorithms learning technique	4	Supervised, Semi supervised, Unsupervised and Reinforcement	Combinatorial but learning process is permutative	Preprocess before training.
17.	Algorithms analysis types	4	Regression, classification, clustering and association	Combinatorial	Regression analysis was applied
18.	Evaluation criteria category	2	Quantitative and Qualitative	Combinatorial	Preprocess before training
19.	Evaluation criteria	9	Stability, efficiency, accuracy, consistency, visualization sharpness, computational complexity. repeatability, straightforwardness and explainability (XAI)	Combinatorial	Preprocess before training
20.	Activation function	24	ReLU. Leaky ReLU, Maxout, Tanh, linear/identity, Binary step, piecewise linear, Sigmoid, Complementary log-log, Bipolar, Bipolar Sigmoid, LeCun's Tanh, Hard Tanh, Absolute, Rectifier, Smooth Rectifier, Logit, Probit, Cosine, Softmax, Maxout, Multiquadratic and Inverse Multiquadratic.		Preprocess before training
21.	Evaluation metrics category	3	Regression, binary classification and multi-class classification	Combinatorial	Regression was the choice of the research
22.	Evaluation metrics	12	Mean Error (ME), Mean Squared Error (MSE), Mean Absolute Error (MAE), Root	Combinatorial	A hybrid approach was considered

List	Variable Name	Internal Features	Description of Internal Features	Category	Remarks
	or Loss functions		Mean Squared Error (RMSE), R Squared, Categorical Cross Entropy, Binary Cross Entropy , Hinge Loss, Squared Hinge, Multi-Class Cross-Entropy Sparse Multiclass Cross-Entropy and Kullback Leibler Divergence.		
23.	Weights	1	Randomly allocated using parameter optimisation techniques	Permutative	Automatically assigned
24.	Bias	1	Automatically selected using libraries.	Permutative	Guided by other factors
25.	Net input	1	Depend on the nature of the input features of the dataset.	Permutative	Guided by other factors
26.	Number of Neurons	1	Determined by a specific mathematical formula	Permutative	Guided by other factors
27.	Number of layers	1	Determined by a specific mathematical formula	Permutative	Guided by other factors
28.	Interconnections	2	Feed-forward and recurrent	Combinatorial and permutative	Guided by other factors
29.	Training process	2	Backpropagation and Backpropagation through time		Automatically implemented through Python libraries.
TOTAL		≥410			

Section C: Summary of Results Produced by Different Best Models

Model	Number of Parameter	Accuracy-Prediction Analysis (GBP_USD Daily Exchange Rate from 1990-2016 by Chang et al.(2018)			Accuracy-Prediction Analysis (JPY_USD Daily Exchange Rate from 1990-2016 by Chang et al.(2018))			Training Efficiency (1)		Stability
		MAE	MSE	R2	MAE	MSE	R2	Time (Seconds)	Efficiency (SelFISA Units)	
Deep LSTM Model influenced by Glenski et. al (2019) and Chalvatzisa et. al (2019)	128513	9.13E-02	2.66E-04	8.89E-01	4.36E-01	2.16E-01	-4.05E+00	4425	29.04248588	125.9021172
Bidirectional LSTMs Model influenced by Sardelicha and Manandhara (2018)	23131	1.67E-02	3.31E-04	9.76E-01	1.72E-01	3.31E-02	2.26E-01	1210	19.11652893	164.599894
Bidirectional GRUs Model influenced by Sardelicha and Manandhara (2018)	17931	5.54E-02	2.36E-03	8.28E-01	3.21E-02	5.61E-03	3.62E-01	963	18.61993769	46.49610679
Attention LSTM Model by Liu (2018)	10276	1.97E-02	2.08E-03	8.85E-01	3.45E-01	1.27E-01	-1.96E+00	3152	3.260152284	178.3931999
SeLFISA Model	**71126**	**1.03E-02**	**2.55E-04**	**9.81E-01**	**1.49E-02**	**3.33E-03**	**4.21E-01**	**7479**	**9.510094932**	**36.50793651**

$$\text{Efficiency (SelFISA Units)} = \frac{\text{Number of parameters}}{\text{Time in seconds}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (1)$$

Stability Based on Percentage Difference =
MAE of GBP_USD Dataset (V1) and JPY_USD Dataset (V2)

$$\frac{|V_1 - V_2|}{\left[\frac{(V_1 + V_2)}{2}\right]} \times 100 \quad \ldots\ldots\ldots\ldots\ldots \quad \ldots\ldots\ldots\ldots\ldots \quad (2)$$

NB: The lower the stability value derived from Equation 2, the more stable the model is.

Section D: Visualisation of Results

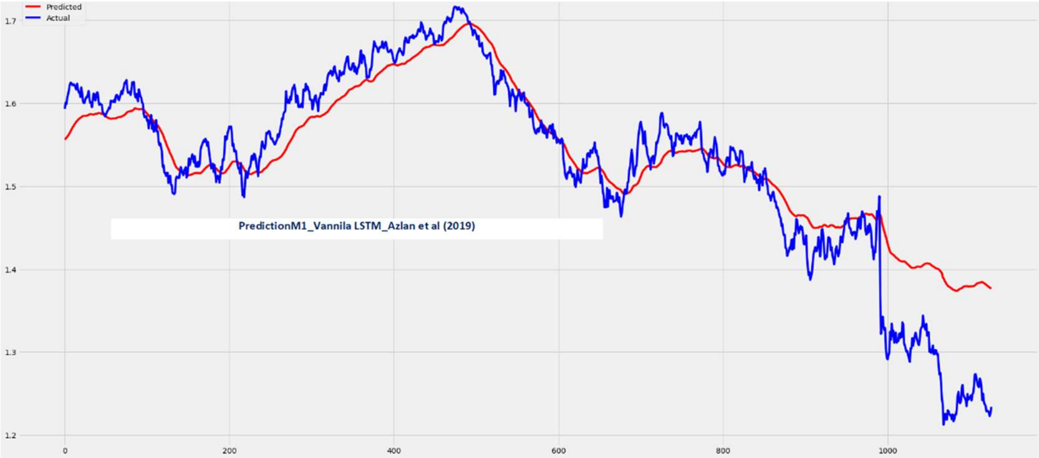

Figure 1- LSTM (32) + Dropout (0.2) + Dense (1) suggested by Azlan et al(2019), Li et. al (2019) and Glenski et. al (2019)

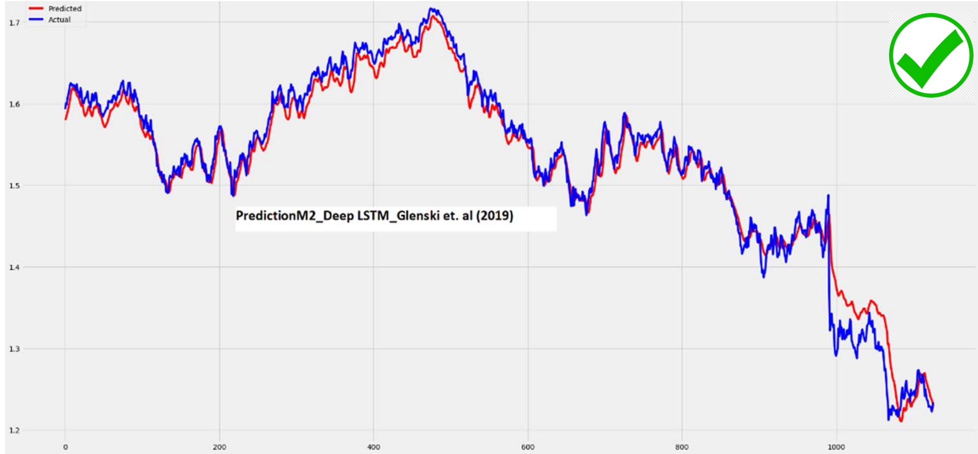

Figure 2- LSTM(32)+LSTM(64)+Dropout(0.2)+LSTM(128)+Dropout(0.5)+Dense(1) by Deep LSTM Model based implemented by Glenski et. al (2019) and Chalvatzisa et. al (2019)

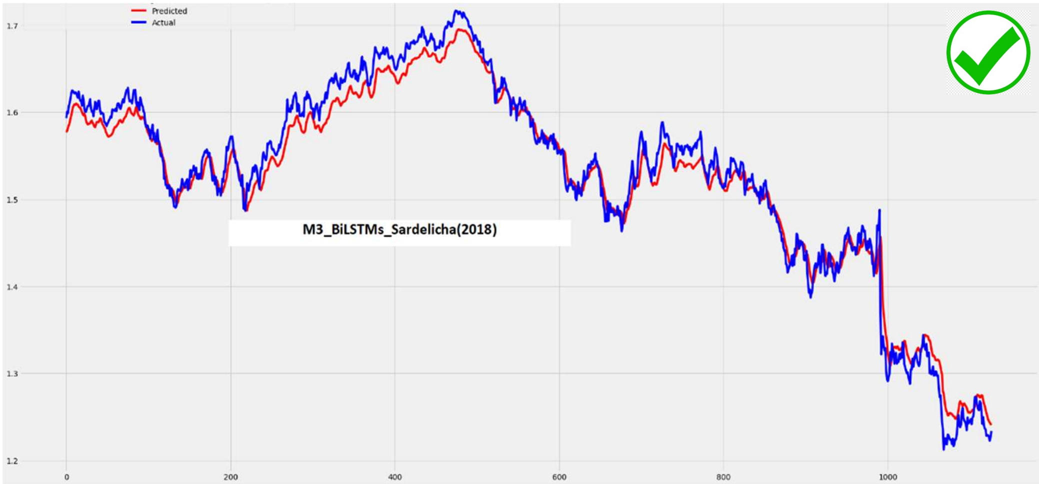

Figure 3- BiD(LSTM(50))+Dense(10)+Dense(10)+Dense(1) influenced by Sardelicha and Manandhara (2018)

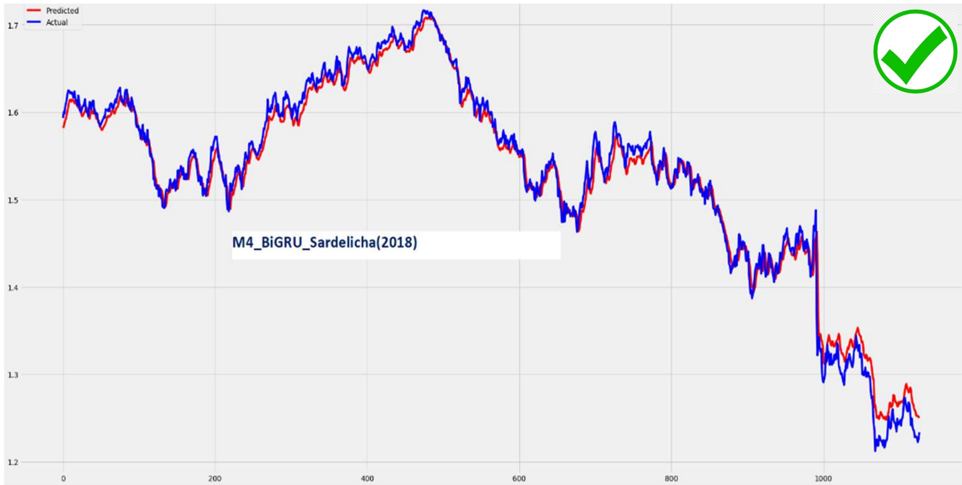

Figure 4 - BiD(GRU(50))+Dense(10)+Dense(10)+Dense(1) by Sardelicha and Manandhara (2018)

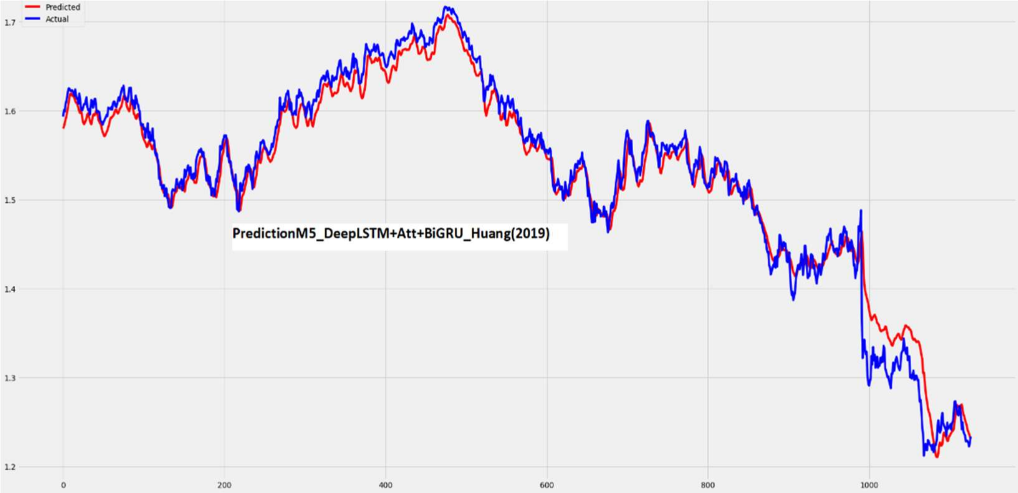

Figure 5 - LSTM(100)+Dropout(100)+Attention(SeqSelfAttention) + LSTM(16)+Dense(10) +Dense(10) + Dense(1) by Deep LSTM by Huang (2019)

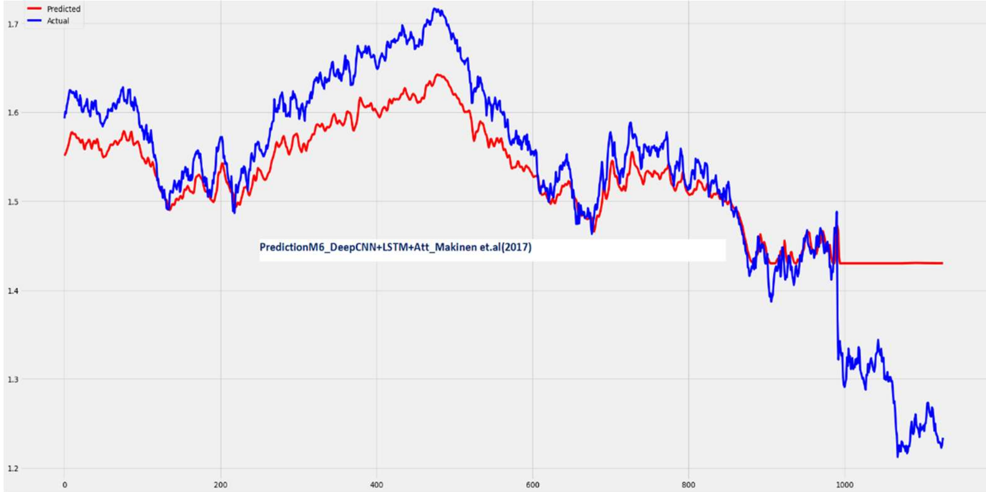

Figure 6 - LSTM(32)+Conv1D(32)+ Dropout(0.2) + Conv1D (16)+Conv1DTr(16)+Dropout(16)+ Conv1DTr(32)+Conv1D (16) + AttSeqSelf(1)+ LSTM(16)+Dropout(0.2)+Dense (1) by Makinen et. al (2017)SeqSelf(1)+LSTM(16)+Dropout(0.2)+Dense (1) by Huang (2019)ttention by Huang (2019)

Figure 7 - LSTM(32)+Dropout(100)+Attention (SeqSelf (32))+LSTM (16)+Dense(10)+Dense(10)+Dense(1) by Liu (2018)

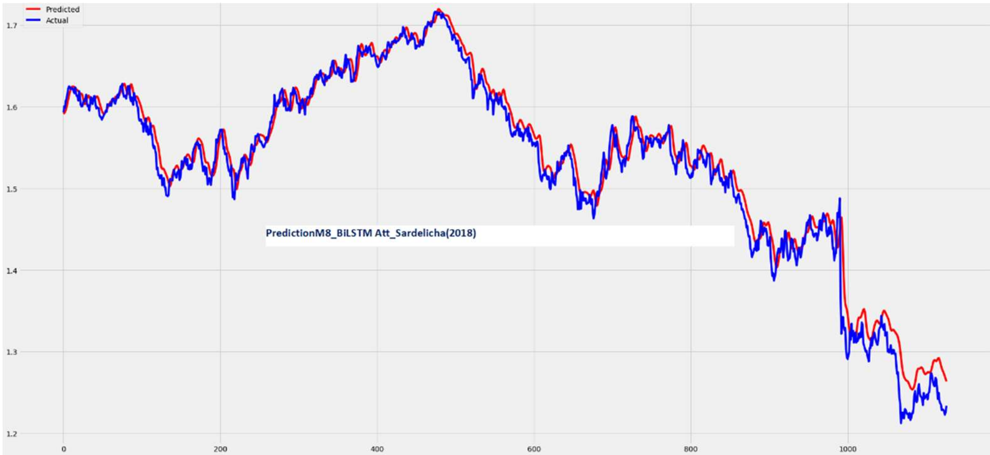

Figure 8 - LSTM(32)+Dropout(0.2)+ Attention (SeqSelf)(32) + Bidirection(LSTM(32))+ Bidirection(LSTM(32)) + Dense(10) + Dense(1) by by Sardelicha and Manandhara (2018)

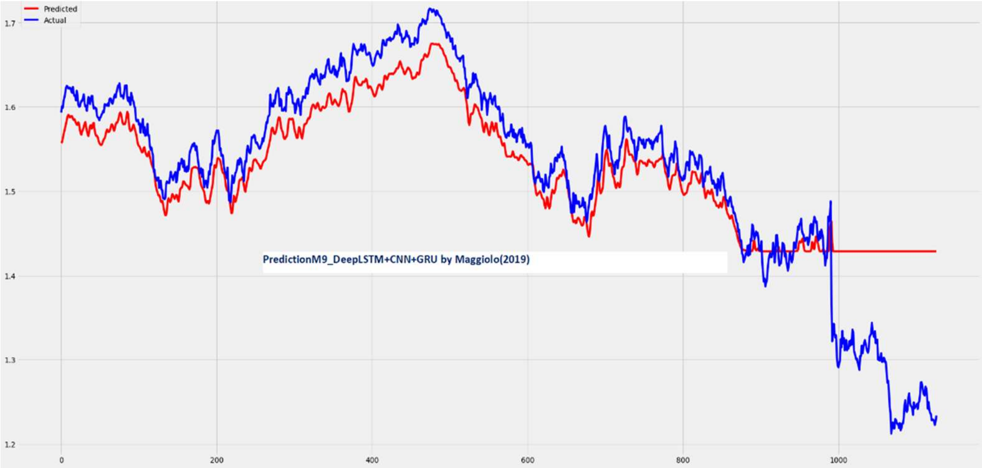

Figure 9 - LSTM(32)+ Conv1D(32)+ Dropout(0.2)+Conv1D(16)+ Conv1DTranspose(16)+ Dropout(0.2)+Conv1DTranspose(32)+Conv1DTranspose(1)+GRU(32)+Dropout(0.5)+Dense(1) by Maggiolo and Spanakis (2019)

Figure 10 - GRU(32)+GRU(64)+Dropout(0.2)+GRU(128)+Dense(1) by GRU by Qin(2019)

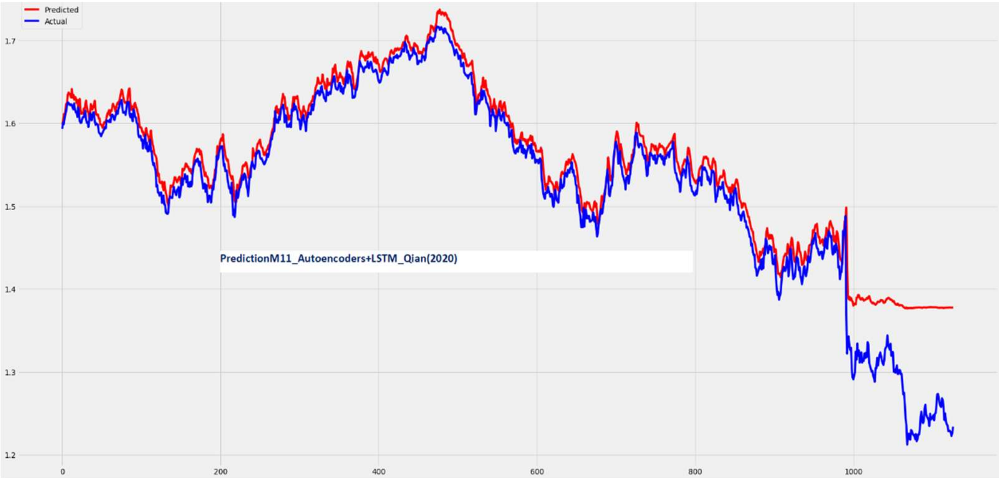

Figure 11- LSTM(32)+LSTM(64)+RepeatVector(64)+LSTM 64)+TimeDist(1)+ LSTM(128)+ Dropout128) + Dense(1 by Qian(2020)

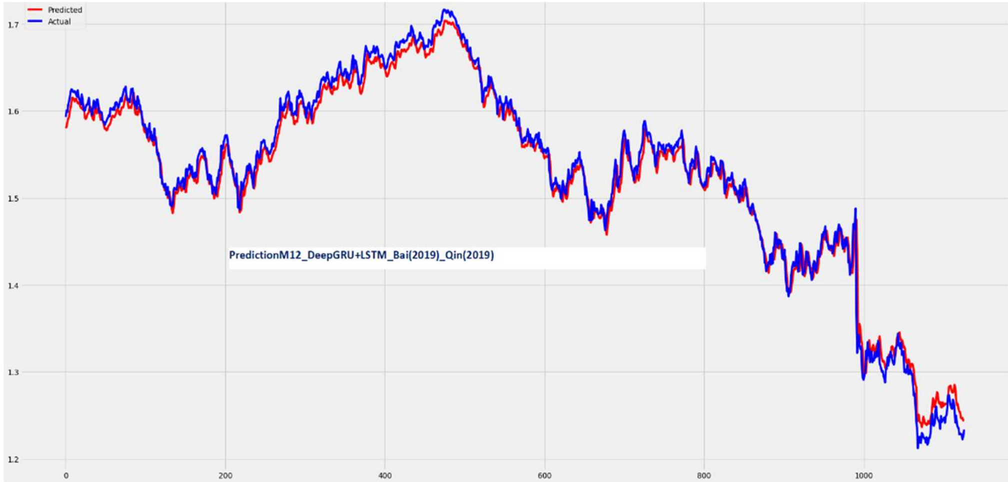

Figure 12 - LSTM(50)+Dropout+LSTM(100)+ Dropout (0.5)+ + GRU(100)+LSTM(100)+ Dropout (0.5)+ +LSTM(100)+ Dropout (0.5)+ Dense(100)+Dense(10)+Dense(10)+Dense(1) by Bai(2019) and Qin(2019)

SeLFISA_Model

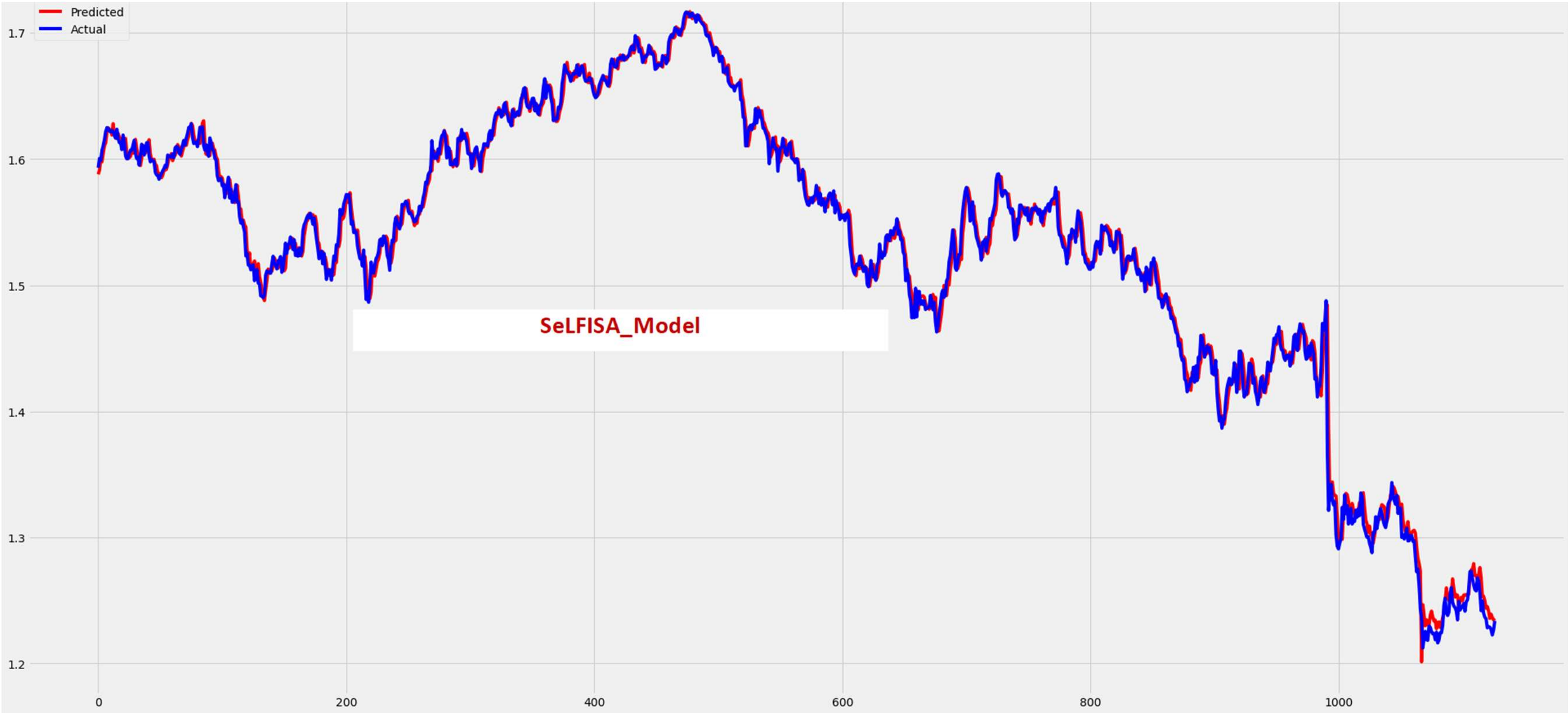

Figure 13 – BiD (GRU(32)) + SeqSelfAtt (att_width=30)+ Dropout(0.2) +BiD (LSTM(32)) +BiD (GRU(32))+ BiD (LSTM(32))+ BiD (GRU(32))+ LSTM (32)+ GRU (32) + Dense(1)