Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματαRESEARCH METHODOLOGY – PART A

EconoPhysics - Financial Forecasts

First semester 2021-22

By Maroulis Adamadios

Professor: Theodoros Karakasidis

**Methods used for forecasting in volatile & unstable markets, ARIMA – LSTM comparison**

Table of Contents

[Abstract 3](#_Toc93416061)

[Keywords 3](#_Toc93416062)

[1 Introduction 4](#_Toc93416063)

[2 SPF comparing ARIMA and XGBoost - LSTM 6](#_Toc93416064)

[2.1 ARIMA model 6](#_Toc93416065)

[2.2 XGBoost and LSTM Models 7](#_Toc93416066)

[2.3 Results 10](#_Toc93416067)

[3 ARIMA AND LSTM COMPARISON IN FORECASTING TIME SERIES 16](#_Toc93416068)

[3.1 Results 19](#_Toc93416069)

[4 ARIMA AND LSTM COMPARISON IN BITCOIN PRICE PREDICTION 20](#_Toc93416070)

[4.1 Dataset 20](#_Toc93416071)

[4.2 ARIMA model 21](#_Toc93416072)

[4.3 LSTM model 22](#_Toc93416073)

[4.4 Results 23](#_Toc93416074)

[5 Conclusion 24](#_Toc93416075)

[REFERENCES 25](#_Toc93416076)

Abstract: The main goal of an investor was always to obtain the biggest return of an investment combined with the lower risk possible, as the years went by there were created needs to invent tools and methods that can minimize that risk and maximize profit either in long or short positions especially in volatile markets. Many of those methods have shown great results in forecasting, better and more efficient, compared to other traditional approaches. Nowadays predicting the direction of stocks, indices, markets can be obtained using various methods through machine and deep learning which gives the ability to collect much more information using time-series analysis to profit from upward or downward trends. In this article we shall discuss some of those methods such as ARIMA and LSTM model and their ability to solve information problems, we will examine what problems can be solved or described better by using the most suited method each time.

Keywords: machine learning, deep learning, stock market, inflation, ARIMA, stock price forecasting, LSTM

# 1 Introduction

Stock-price forecasting is one of the most attractive and difficult areas for research in quantitative investing time-series analysis just because it is essential for investment decision-making, managing a portfolio and its risk. There have been made great efforts through analysts, researchers and practitioners to understand and comprehend the price dynamics (Eugene F. Fama 2017).

Stock market predictions remains to be one of the most challenging problems in the modern financial world, researchers used to choose more traditional methods to comprehend and analyze datasets. ARIMA model have shown good results for machine learning (ML), although it seemed to provide good results in the past few years studies have shown that using machine learning (ML) and deep learning (DL) methods provide much more accurate results and better understanding in terms of information especially in bigger datasets and have obtained significant developments in various domains (Liu, Pun 2021). Those methods were used to capture time-series features, variations of those methods were used too such as seasonal ARIMA and ARIMA with explanatory variables in order to make different approaches in those kinds of problems. As we know from the literature of Berradi, Lazaar et. al. those methods can be helpful for short-period predictions, however they only use methods that are based on regression while they can be applied only to problems that are linear, therefore there is much less effectiveness for long-term predictions.

Another approach is deep and machine learning that have been proposed as an improvement to stock price forecasting which includes ANNs\*, CNNs\* and RNNs\* such as LSTM (Vuong, Dat et. al. 2021) which will be examined further in the following chapters.

Using time series models for data analysis can be very complex as there are many elements that can influence the behavior of a stock price (e.g., inflation, societal behaviors, seasonality, volatility, sentiment, economic policy).

This study, therefore, will present machine learning and deep learning analysis used for forecasting that have been proposed based on advanced methodologies providing evidence that perform better than more traditional approaches in volatile and unstable markets, this study will try to examine if deep learning approaches does indeed perform better in some cases.

In chapter 2 we shall examine in depth an approach based on both machine and deep learning techniques in order to strengthen SPF performance. The method presented by P. H. Vuong, Trinh Tan Dat et. al. provides very good results to both stock and Forex data when applied and showed that their approach based on XGBoost and LSTM outperformed the (ARIMA) approach when it comes to MAE\*, MSE\* and RMSE\*.

Furthermore, in chapter 3 ARIMA and LSTM will be examined and compared in forecasting time series, it will be presented to examine how the results differ when comparing a deep learning-based approach to a traditional based one. We shall see that the LSTM model will outperform the ARIMA model on the average reduction in error rates 84 – 87 percent indicating the superiority of a deep learning approach (Siami-Namini S., Tavakoli N. et. al. (2019)). It’s always a challenge to propose a model that outperforms such successful techniques in a dataset of major indices.

Lastly, in chapter 4 we shall examine how those two models (ARIMA, LSTM) performed when trying to predict the well-known digital currency Bitcoin. The author concluded that both models worked fine with the ARIMA model being a time-saver compared to LSTM. In addition, the author advocated that LSTM could be much more efficient in terms of prediction and precision rate when it comes to larger periods of time. (Yiqing Hua 2020)

# 2 SPF comparing ARIMA and XGBoost - LSTM

# 2.1 ARIMA model

Vuong, Dat, et. al. proposes a scheme using an ARIMA model as the baseline to be compared with their approach, as it is widely used in the field for time-series forecasting (Aasi, Imtiaz 2021). ARIMA is basically model of ARMA combining AR process (autoregressive) and MA process (moving average), I stand for integrated and makes the time series constant. The model includes the following processes: AR and MA, using a constant and assuming to be a Gaussian white noise series having mean=0 and var: ( > 0). The model is given by the following function, being of the order (p,d,q):

= +++ . . . ++++ . . . + (1) *(*Vuong, Dat 2021)

Where θ is constant; ≠ 0 representing coefficients of autocorrelation at lag:1, . ., p (p designates AR order), while ≠ 0, j=0, q are weighted coefficients that are used from both current and prior values of a stochastic term in the time series (q designates MA order). As the authors refer to, the ARIMA model is able to deal with nonconstant time series based on its integrated component as it evolves differencing which is applied to form the nonconstant time series constant. Variable d is used to measure the difference in the observations at given times.

Carefully and effectively selecting the variables d, p and q makes the ARIMA model reliable so the authors determined the variables (p,q) using ACF\*, PACF\* and other indicators like Log-L\*, AIC\* and BIC\* all widely known for their use case, for the parameter d, they used the Dickey-Fuller test.

# 2.2 XGBoost and LSTM Models

Firstly, the authors considered the extreme gradient boosting (XGBoost) for selecting the features in order to separate the valuable features to predict, using high-length time-series data as well as rejecting unnecessary ones. Those features were fed into the LTSM model in order to forecast stock prices. We present a quote **Fig. 1** from Vuong, Dat (2021) which shows a diagram of the method.

Diagram

Description automatically generated

**Figure 1:** Diagram of the SPF approach *(*Vuong, Dat 2021)

The authors refer to Chen, Zhang and Zhang, Lu to describe XGBoost as a powerfull machine learning algorithm that is used for structured or arranged in tabular form data, they also point out that it can enhance the speed and the performance based on the execution of gradient-boosted decision trees. In addition, they note that it is used worldwide for feature selection because of its high scalability, parallelization, speed and efficiency.

Considering N observations dataset, D={X,Y}, with X={, ( 2 Ꞓ) and Y={,(Y Ꞓ R) designating the training features and the observed value/target, assuming K numbers of gradient-boosting iterations and M additive functions are used to predict the output. is designating the prediction value of the ἱth feature vector at the ꬺth boost, , as an independent true structure, q, with leaf wight ω ( representing the score on the jth leaf in the tree).

With a given input feature vector, the authors made the final prediction by gathering up the values through all leaves based on:

*(*Vuong, Dat 2021) (2)

Where F = {} ( q: T, ω ) symbolize the space of regression trees, q declares the structure of each tree which corresponds to an entry in the analogous leaf index, and T stands for the number of leaves in the tree. The main idea is to minimize loss function:

*(*Vuong, Dat 2021) (3)

Where L is the subtract of prediction and target . To be able to deal with the model’s adaption rate for the training dataset the authors added the learning rate of shrinkage factor. Furthermore, a penalty factor was added, Ω() that makes the model simpler to the objective function in Eq.(3), the generalized objective function of XGBoost is:

*(*Vuong, Dat 2021) (4)

Features that were selected from XGBoost were fed into the LSTM model. While LSTM is an expansion of RNN it lowers the effect of the vanishing gradient problem.

Fig.2 presents a standard LSTM unit structure for calculating cells, a basic LSTM unit consists of a memory cell, an input, an output and a forget gate. Both future and prior information are equally important. The first two gates are used to store and gather information in the cell for big periods, while the input gate is responsible for adding new information and the output gate is responsible for deciding what part of the memory provides for the output. Lastly, the forget gate is used for clearing the memory in the cell, since this gate is responsible for deciding which information is useful and which should not be used from memory, it correctly shows the long-term dependence that appears in time series *(*Vuong, Dat 2021). In short, LSTM can be described as a kind of Recurrent Neural Network (RNN) with the ability to store values from previous stages in order to use them in the future. (Siami-Namini, Tavakoli 2019)

Diagram

Description automatically generated

**Figure 2:** LSTM unit *(*Vuong, Dat 2021)

With a frame of the sequence of features = ,.. , every time gets inside the sequence from LSTM the hidden state is updated with a nonlinear function that takes both current input and previous state . In more detail, for frame at current state, is the hidden state at previous state t-1, while is the cell state of the memory at the previous state (t-1). The unit starts by calculating the forget gate , the input gate , the output gate , and the candidate context as follows *(*Vuong, Dat 2021):

*=([; ] +),* (5)

*=([; ] +),* (6)

*=([; ] +),* (7)

*=([; ] +),* (8)

Here, the author describes the variables W and b as the weight matrices and bias vector parameters, that need to be learned in order to train the model.

Parameter is a sigmoid function while is a hyperbolic tangent function.

Furthermore, the cell state and hidden state at current time t are estimated as follows:

*=ο +ο ,* (9)

*= ο* (10)

Where o indicates the Hadamard product (element-wise product), and designate the hyperbolic tangent function. The model itself is designating the direction while it is used to indicate that the input time series is temporary. In addition, it takes full advantage of future contextual information. *(*Vuong, Dat 2021)

Furthermore, we present Figure 3 that indicates the block diagram of the LSTM model for SPF showing that it is a much more complicated model when having more layers*:Text

Description automatically generated with low confidence*

***Figure 3:*** *Diagram of LSTM model for SPF (*Vuong, Dat 2021)

# 2.3 Results

This proposed method was evaluated using data that was collected from the Forex market (“2019 International Data Science Competition,”), containing information from 01/01/2008 to 03/19/2018 and having 709.314 total observations *(*Vuong, Dat 2021). The author provides table 1 showing a synopsis of the statistics of the Forex market, using for prediction target only closing prices. To estimate how ARIMA performed, the author considered a subset of almost 60,000 observations with the 1-hour price as the main dataset was used to evaluate the performance of LSTM. *(*Vuong, Dat 2021).

Table

Description automatically generated

*(*Vuong, Dat 2021)

To analyze the ARIMA model the sub dataset was separated randomly into two parts, 70% for training and 30% for testing, while 41,365 observations trained the model and 17,729 tested it. To be able to estimate the most suited (p, d, q) for the model the training data were used while the augmented Dickey-Fuller test was used to estimate d and it was discovered that the observations were constant at d=1.

Furthermore, we present table 2 and 3 in which ACF\* and PACF\* are estimated using train data’s prices and the results of various ARIMA variables for the Forex market. The model was chosen based on criteria such as AIC\*, BIC\* and Log-L\*, which indicated that the best model was ARIMA (0,1,1). We can see that after the second lag both ACF and PACF are very close to zero *(*Vuong, Dat 2021).

Table

Description automatically generated

*(*Vuong, Dat 2021)

*Table

Description automatically generated*

*(*Vuong, Dat 2021)

In the XGBoost and LSTM approaches, the dataset was randomly separated in three parts, 60% was used for training the model while the rest was equally separated for validation and testing.

As P. H. Vuong, Trinh Tan Dat, et. al. mentions in bio almost 400,000 observations were used to train the model, close to 170,000 for validation and almost 140,000 for testing all high-dimensional data containing two hundred features. XGBoost was used to pick the most important features using the F-score prices while Adam optimization was used as an optimizer and 50 epochs were used to train the LSTM model in Keras. The outcome was 10 important features that were fed into the LSTM model for best performance. Figure 4 presents important features selected based on XGBoost. Lastly, MAE\*, MSE\* and RMSE\* were used to estimate the efficiency of the system with smaller values indicating efficiency of the system *(*Vuong, Dat 2021).

Table

Description automatically generated with medium confidence

**Figure 4:** Feature importance *(*Vuong, Dat 2021)

P. H. Vuong, Trinh Tan Dat, et. al. advocated that both approaches proved to have great prediction efficiency with the second one making a little bit better performance in achieving the best accuracy and being promising for long-term time-series prediction as it has the privilege of choosing important and relevant information in comparison to the ARIMA model which lacks the memory privilege. Below we present tables 4, 5 and 6 in which we show the prediction results of the models and a comparison of them and their performance. The authors in their own words describe their proposed approach as a great method to improve the efficiency of SPF. *(*Vuong, Dat 2021)

Table

Description automatically generated

*(*Vuong, Dat 2021)

Table

Description automatically generated

*(*Vuong, Dat 2021)

*Table

Description automatically generated*

*(*Vuong, Dat 2021)

# 3 ARIMA AND LSTM COMPARISON IN FORECASTING TIME SERIES

An interesting question was asked by Siami-Namini S., Tavakoli N. et. al. as the authors presented a paper regarding how accurate and precise traditional models are in comparison to algorithms that use deep learning. ARIMA model was chosen from the authors because of the nonconstant property of the data gathered and modeled while the LSTM model was chosen for its use in maintaining and training the features of given data for a bigger period. The authors compared those models in their ability to reduce error rates using a data set which is presented in table 7, the data were extracted from Yahoo finance website, the Federal Reserve Bank of St. Louis and the International Monetary Fund (IMF) (Siami-Namini, Tavakoli 2019).

**Table 7:** Historical / economical monthly financial time series

Table

Description automatically generated

(Siami-Namini, Tavakoli 2019)

Table

Description automatically generated

(Siami-Namini, Tavakoli 2019)

**Table 8:** Observations of Indices

Table

Description automatically generated

(Siami-Namini, Tavakoli 2019)

Table 8 lists the observations which were split in two parts for each dataset in which 70% were used for training and 30% for testing. RMSE\* was considered to estimate the efficiency of forecasting that was managed by the model (Siami-Namini S., Tavakoli N. et. al. (2019)).

*(11)*

The authors used a programming language called Python to be able to implement the algorithms along with Keras, an open source neural network library and Theano, a numerical computation library Siami-Namini, Tavakoli (2019), J. Brownlee (2016).

*Text

Description automatically generatedText, letter

Description automatically generated*

(Siami-Namini, Tavakoli 2019)

Listing 1 shows the ARIMA algorithm that was used while Listing 2 shows the LSTM algorithm. We should note that for the LSTM algorithm, Keras library along with Theano were used on a cluster of high-performance computing center. (Siami-Namini, Tavakoli 2019)

# 3.1 Results

In table 9 we can see the amazing results that the Rolling LSTM model brings to the light in comparison to the Rolling ARIMA model with the first one outperforming the second one at a reduction in error rates of over 80%. The financial time series related data show that the RMSE using the models are close to 510,000 and 65,000, estimating an average of almost 87.500 reductions in error rates by LSTM. In addition, data shows a reduction of 84.394 in RMSE with an average estimation value of 5.999 in ARIMA and 0.936 in LSTM. More specifically, the data show that the LSTM-based algorithm in comparison to the ARIMA algorithm made almost 85% better prediction on average. (Siami-Namini, Tavakoli 2019)

**Table 9:** The RMSEs of ARIMA and LSTM models

Table

Description automatically generated

(Siami-Namini, Tavakoli 2019)

# 4 ARIMA AND LSTM COMPARISON IN BITCOIN PRICE PREDICTION

In this chapter we shall examine another interesting approach regarding a special type of digital currency which is becoming more and more widespread in our days, broadly used in many online trading systems, some regard it to have the value of gold in a digital form.

In the last decade its price has went through series of fluctuation and at the time written by the author its price was about 7000 from BTC to USD (Yiqing Hua 2020). As of today, its price has peaked at about 69000 from BTC to USD and at the time its currently valued at about 45000 from BTC to USD. (Yiqing Hua 2020).

The goal was a comparison in accuracy of bitcoin price prediction using two different models, ARIMA and LSTM while using Pycurl, a very common python library for collecting online data, using real-time price data from Bitfinex.

# 4.1 Dataset

To obtain a dataset which can be trained, Yiqing Hua collected data from the API (application that allows real-time price info.) and used five seconds data from the website BitFinex. Ten thousand prices information were collected and separated into two parts 80% for training and 20% for testing. Table 10 shows the fundamental attributes of the information of BTC/US (Yiqing Hua 2020).

**Table 10:** Dataset attributes

Table

Description automatically generated

(Yiqing Hua 2020)

# 4.2 ARIMA model

After pre-working on the data, the time-series was estimated to 733 points in training set and 100 points in testing set and the model chosen for ARIMA was (1,1,0) which yielded the following prediction equation:

(12)

(Yiqing Hua 2020)

Figure 5 presents the forecast of the next 1 step (a) and the forecast of the next 5 steps (b) based on the previous price.

Chart, line chart, histogram

Description automatically generated

Fig.5 Forecasting with ARIMA

(Yiqing Hua 2020)

# 4.3 LSTM model

The author trained the model in 100 epochs containing 10 pieces of continuous data in every round and the findings are shown in Fig.6 were the loss plot tracked from every epoch an immediate loss after the first one closing to 0.02 and after the fifth being close to 3 x (Yiqing Hua 2020).

*Chart

Description automatically generated*

*Fig.6 Training Loss* (Yiqing Hua 2020)

The model performed a decent prediction in the trained LSTM model while the average Error rate was found to be 0.4765938, with a standard deviation of 2.092208. Five and ten previous data models were also tried in testing data and compared to single previous data it was found to be worse than expected as it presented a negative effect even when trying to capture the fluctuation of data all of which are presented in figure 7 below. (Yiqing Hua 2020).

*Chart, histogram

Description automatically generated*

*Chart

Description automatically generated*

*Fig.7* LSTM prediction result and prediction with different strategyYiqing Hua (2020)

# 4.4 Results

The author advocated that both models could handle the job well enough, the LSTM model would be a much more time consumer compared to the ARIMA model but on the other hand the LSTM model could be much more efficient in its predictions and in its precision rate. The outcome seems to be that ARIMA can perform very well in smaller periods of time but as the time goes by the efficiency would fall while the LSTM can lead to better results when taking less previous data under consideration to make predictions and can handle better larger periods of time with a higher precision rate.

# 5 Conclusion

This study examined three proposed methods in which two different methods are compared, in the first case XGBoost was used as the feature-selection method and LSTM was compared to an ARIMA model in terms of better performance in stock price forecasting, the results shown indicate that the deep learning method performed much better as far as the SPF system when compared to the ARIMA model as it has the advantage of selecting the most useful information while the ARIMA model lacks the memory privilege. It is important to note that the authors themselves described their approach as a great method to improve the efficiency of SPF system.

In the second case, we examined the results of a rolling LSTM-ARIMA comparison in which the authors provided experimental evidence that the LSTM model outperformed the ARIMA model by over 80% reduction in error rates. More specifically, data showed that the LSTM-based algorithm in comparison to the ARIMA algorithm made almost 85% better prediction on average.

Lastly, in the third case we examined the Bitcoin price prediction case study in which both ARIMA and LSTM models were efficient with the second one being more time-consuming compared to the first one. In addition, the LSTM model could be much more efficient in terms of prediction and precision rate as far as the estimated period of time, while the ARIMA model proved to be an excellent competitor when it comes to a small period of time.

The goal of this work is to provide evidence and contribute to the argument that as the technology advances, better and better tools, models, theories seem to appear and in many cases, they are more efficient and perform better than more traditional tools in many different sectors and especially in the economic field that we are trying to direct our study.

The work presented in this paper support the benefits of applying deep learning techniques in comparison to more traditional techniques, although more research is needed to conclude that hypothesis as there are several other models that perform better when compared to deep learning approaches in many cases.

# REFERENCES

|  |
| --- |
| [1] [Aasi, B.](https://www.scopus.com/authid/detail.uri?origin=resultslist&authorId=57211404896&zone=), [Imtiaz, S.A.](https://www.scopus.com/authid/detail.uri?origin=resultslist&authorId=57223999638&zone=), [Qadeer, H.A.](https://www.scopus.com/authid/detail.uri?origin=resultslist&authorId=57224013422&zone=), [Singarajah, M.](https://www.scopus.com/authid/detail.uri?origin=resultslist&authorId=57224001661&zone=" \o "Show author details), [Kashef, R.](https://www.scopus.com/authid/detail.uri?origin=resultslist&authorId=23980235900&zone=" \o "Show author details), 2021 IEEE International IOT, Electronics and Mechatronics Conference, IEMTRONICS 2021 – Proceedings 9422526 |
| [2] [Berradi, Z.](https://www.scopus.com/authid/detail.uri?origin=resultslist&authorId=57207622928&zone=), [Lazaar, M.](https://www.scopus.com/authid/detail.uri?origin=resultslist&authorId=57200595982&zone=" \o "Show author details), [Omara, H.](https://www.scopus.com/authid/detail.uri?origin=resultslist&authorId=57195491588&zone=" \o "Show author details), [Mahboub, O.](https://www.scopus.com/authid/detail.uri?origin=resultslist&authorId=23969225400&zone=" \o "Show author details) 2020 [IAENG International Journal of Computer Science](https://www.scopus.com/sourceid/15900154752?origin=resultslist) 47(3), pp. 436-441 |
| [3] Forex market dataset, “2019 International Data Science Competition,” 2019. [Online]. Available at: https://www. isods.org/news-times/item/1-2019-international-data-science-competition |
| [4] [Fama, E.F.](https://www.scopus.com/authid/detail.uri?origin=resultslist&authorId=6603950815&zone=), [French, K.R.](https://www.scopus.com/authid/detail.uri?origin=resultslist&authorId=7006094907&zone=) 2017 [Journal of Financial Economics](https://www.scopus.com/sourceid/24379?origin=resultslist) 123(3), pp. 441-463 |
| [5] <https://www.bitfinex.com/> |
| [6] <https://finance.yahoo.com/> |
| [7] Hua Y. Bitcoin price prediction using ARIMA and LSTM[C]//E3S Web of Conferences. EDP Sciences, 2020, 218 |
| [8] J. Brownlee, “Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras,” https://machinelearningmastery.com/ time-series-prediction-lstm-recurrent-neural-networks-python-keras/, 2016. |
| [9] [Liu, R.](https://www.scopus.com/authid/detail.uri?origin=resultslist&authorId=57226005456&zone=), [Pun, C.S.](https://www.scopus.com/authid/detail.uri?origin=resultslist&authorId=55623304300&zone=) 2021 [Journal of Banking and Finance](https://www.scopus.com/sourceid/17472?origin=resultslist) 136,106416 |
| [10] Sima Siami-Namini, Neda Tavakoli, Akbar Siami Namin ‘’ A Comparison of ARIMA and LSTM in Forecasting Time Series’’ (2019) 2018 17th IEEE International Conference on Machine Learning and Applications |
| [11] [Chen, W.](https://www.scopus.com/authid/detail.uri?origin=resultslist&authorId=57212625483&zone=), [Zhang, H.](https://www.scopus.com/authid/detail.uri?origin=resultslist&authorId=57220771540&zone=), [Mehlawat, M.K.](https://www.scopus.com/authid/detail.uri?origin=resultslist&authorId=23035630900&zone=" \o "Show author details), [Jia, L.](https://www.scopus.com/authid/detail.uri?origin=resultslist&authorId=57192080197&zone=) 2021[Applied Soft Computing](https://www.scopus.com/sourceid/18136?origin=resultslist) 100,106943 |
| [12]T. Kaczmarek, [Będowska-Sójka,](https://www.scopus.com/authid/detail.uri?origin=resultslist&authorId=56893754600&zone=" \o "Show author details) [Grobelny, P.](https://www.scopus.com/authid/detail.uri?origin=resultslist&authorId=57194383683&zone=), [Perez, K.](https://www.scopus.com/authid/detail.uri?origin=resultslist&authorId=57221374704&zone=) 2022 [Research in International Business and Finance](https://www.scopus.com/sourceid/12056?origin=resultslist) 60,101610 |
| [13] Vuong, P.H., Dat, T.T., Mai, T.K., Uyen, P.H., Bao, P.T. ‘’2021’’ Computer Systems Science and Engineering 40(1), pp.237-246. |
| [14] Y. Zhang, [Lu, S.](https://www.scopus.com/authid/detail.uri?origin=resultslist&authorId=57226706311&zone=) 2021 [ACM International Conference Proceeding Series](https://www.scopus.com/sourceid/11600154611?origin=resultslist) pp. 58-64 |

[\*]

Table

Description automatically generated