Norbert Błaszczyk
Opole University of Technology, Computer Science,
ul. Prószkowska 76, Poland, Opole, 45-758,
norbert.blaszczyk@student.po.edu.pl,
Adrian Michalik
Opole University of Technology, Computer Science,
ul. Prószkowska 76, Poland, Opole, 45-758,
adrian.michalik@student.po.edu.pl

Using GMM-HMM (Gaussian Mixture Model) to predict the state of the cryptocurrency market based on historical data

KEYWORDS
HMM, GMM, cryptocurrency, prediction

ABSTRACT

In recent years, the popularity of the cryptocurrency market has increased; two examples of well-known cryptocurrencies are Bitcoin and Ethereum. The Aim of this paper is to examine the (GMM-HMM) Gaussian Mixture Model-Hidden Markov Model 's potential for forecasting the bitcoin market. Effective forecasting is essential for making well-informed investing decisions given the volatility and uncertainty of the market. The GMM-HMM model can be used to forecast the bitcoin market because it can take into account stochastic elements and ambiguity as well as examine several possible outcomes. The effectiveness of the GMM-HMM model will be assessed along with its use in predicting future bitcoin price values and market trends. In order to learn more about prices, sales volume, market capitalization, and other potential market influences for bitcoin, historical market data will be examined. Statistical and correlation analysis will then be performed to look for any relevant relationships between these factors. The Prophet library's forecasts were more accurate with a smaller variation from the actual exchange rate than the GMM-HMM model's, which had predictions that were not quite as accurate. By incorporating other variables into the model, such as news sentiment analysis, or by experimenting with different time series forecasting methods, the existing strategy could still be made better. This study attempted to predict bitcoin exchange rates using a GMM-HMM technique, however it was unsuccessful. The study also emphasized the difficulties in predicting the cryptocurrency market and offered ideas for potential improvements, including as the addition of more factors and the investigation of other forecasting methods.

1. INTRODUCTION

The cryptocurrency industry has become more well-known and active over the past few years. Bitcoin and Ethereum are two examples of the cryptocurrencies that have gained popularity as investment vehicles and payment methods. More people are beginning to invest in cryptocurrencies as a result of the market's growing popularity [1, 2]. The cryptocurrency market, however, is unstable [3]. Frequently, unanticipated occurrences like the enactment of new rules by governments or hacking are to blame for the volatility of cryptocurrency values. Effective bitcoin market forecasting becomes essential in this situation for making informed investment decisions. For investors and financial experts, predicting the state of the cryptocurrency market is crucial since it enables them to make informed investment choices. Effective bitcoin market forecasting enables investors to better comprehend market trends and steer clear of risky investments [4]. This article's goal is to use past data to anticipate the state of the cryptocurrency market using the Gaussian Mixture Model (GMM-HMM). A mathematical model called GMM-HMM in which Gaussian Mixture Model to predict next move based on previous moves and Hidden Markov Model to derive a general solution from demonstrations [5] enables system state forecasting by predicting future values of previous data. The cryptocurrency market, which is uncertain, volatile, and challenging to anticipate, benefits greatly from this strategy. The GMM-HMM model is very helpful for predicting the bitcoin market since it allows for the incorporation of stochastic components and ambiguity. The ability to analyze multiple potential outcomes is crucial for the bitcoin market, which lacks stability and historical data.

Therefore, the goal of this paper is to evaluate the GMM-HMM model's performance in predicting future market trends and to apply the model to estimate future cryptocurrency price values. There are numerous publications on bitcoin market predictions like in [6–7], GMM-HMM models are rarely used. Given that GMM-HMM is one of the machine learning models [8, 9], it is important to pay attention to studies that focus on using just machine learning to forecast the cryptocurrency market. It is also recommended to read papers on the use of GMM-HMM in forecasting financial markets [10, 11], as they can teach you a lot about the strengths and weaknesses of this model. We will make use of market historical data to respond to the study questions. We will gather information on costs, sales volume, market capitalization, and other elements that could influence the bitcoin market [12]. After that, we will statistically analyze the data to learn more about its distribution, trends, and seasonality. In order to uncover potential connections between various elements, we will also perform correlation analysis. The GMM-HMM model will then be used to project future values for the historical data [13]. Finally, in order to assess the model's performance, we will contrast its forecasts with the actual values of the cryptocurrency market. To find potential model issues and suggest fixes, we will also perform forecast error analysis.

2. MATERIALS AND METHODS

The cryptocurrency price measurements were carried out using the https://minapi.cryptocompare.com/data/v2/histodayAPI for the period from 12/07/2017 to 02/01/2023 (approx. 2000 days). The API was used to retrieve data from the website and make it available for analysis. The data used for the analysis are shown in Table 1. The measurement system consists of a PC with a Python script installed and the data-retrieval API available at https://min-api.cryptocompare.com/-data/v2/histoday. A Python script that made use of the pandas, numpy, matplotlib, hmmlearn, prophet, and requests libraries was used to analyze the historical cryptocurrency data. These packages include tools for charting, data processing, and API access. The Analysis consisted of training an GMM-HMM model on the data, and using the model to make a prediction for next month.

Table 1. Dataset used for the analysis

Date	Open	High	Low	Close	Volume	Market Cap
2017-07-20	2282.58	2353.56	2345.73	2232.73	313626.43	807614840.88
2017-07-21	2835.02	2734.53	2634.53	2356.62	310335.43	807613890.88
2017-07-22	2675.08	2723.52	2453.01	2575.63	310335.43	807316489.88
…	…	…	…	…	…	…
2023-01-10	17179.03	17179.03	17179.03	17179.03	310335.43	807416890.88
2023-01-11	17619.02	17359.02	15689.02	13699.02	330455.43	807416693.88

3. POSSIBLE APROACHES

The calculation methods such as: ARIMA, LSTM, STL, Prophet, and GMMHMM are all different types of time series forecasting methods. They are not considered deep learning models (Tab. 2). They are traditional statistical models used for time series forecasting. GMMHMM is a type of Hidden Markov Model (HMM) which is a statistical model and not considered a deep learning model. Deep learning models, such as Long Short-Term Memory (LSTM) networks, are a type of neural network that are commonly used for time series forecasting. However, LSTM is also used in traditional methods, and it is not necessary to use deep learning for time series forecasting, it depends on the problem, data and the specific use case.

The data can be analyzed using the following techniques: A linear model called ARIMA (Auto-Regressive Integrated Moving Average) is frequently employed for time series forecasting. To make predictions, it combines moving average and autoregression models. Recurrent neural networks of the LSTM (Long Short-Term Memory) variety are particularly effective for time series data. It is helpful for modeling data sequences over time because it can capture long-term dependencies in the data. Facebook created a library called Prophet with the express purpose of time series forecasting. To make predictions, it combines Bayesian inference with additive models. Seasonal Decomposition of Time Series (STL) is a technique that decomposes a time series into its seasonal, trend, and

residual components. This makes it possible to identify patterns in the data and make predictions based on them. GMM is a variant of HMM where the observation probability density function is modeled as a Gaussian Mixture Model (GMM) instead of a discrete probability distribution. This allows for modeling continuous observations such as stock prices, speech signals, etc.

Table 2. Previous studies and their approaches to the problem of predicting the state of the markets

Reference	Approach	Data source
1. Predicting stock market index using LSTM	LSTM	US stock market index
2. Classifying trend movements in the MSCI U.S.A. capital market index—A comparison of regression, arima and neural network methods	ARIMA	Capital market index
3. Asian stock markets closing index forecast based on secondary decomposition, multi-factor analysis and attention-based LSTM model	LSTM	Asian stock markets
4. Integrating Navier-Stokes equation and neoteric iForest-BorutaShap-Facebook's prophet framework for stock market prediction: An application in Indian context	Prophet	Indian stock markets
5. Seasonal forecasting of agricultural commodity price using a hybrid STL and ELM method: Evidence from the vegetable market in China	STL	Vegetable market in Chin
6. Bank stocks inform higher growth—A System GMM analysis of tenemerging markets in Asia	GMM	markets in Asia

4. CALCULATION ALGORITHMS

GMM-HMM (Gaussian Mixture Model Hidden Markov Model) approach was used in this study to improve the accuracy of predictions compared to a simple HMM. The simple HMM approach used discrete ranges for the observation (*close – open/open*). However, the GMMHMM approach uses continuous observations, which allows for more accurate predictions. The hmmlearn library, which has GMMHMM capabilities, was used to train the models. The observations used to train the model were vector observations, given by the formula:

$$O_t = \left(\frac{close-open}{open}, \frac{high-open}{open}, \frac{open-low}{open}\right)$$

representing the fractional change in close, high, and low prices respectively. The Maximum a Posteriori (MAP) algorithm was used to train the model with different number of states and different number of mixtures. For prediction, the possible observations were defined as the following 3 (Table 3).

Table 3. Possible observations

Observation	Range	Point num.
$\frac{close-open}{open}$	-0.1 to 0.1	50
$\frac{high-open}{open}$	0.0 to 0.1	10
$\frac{open-low}{open}$	0.0 to 0.1	10

The MAP algorithm [14] is a Bayesian approach, where the likelihood of the observations is maximized given the model's parameters. It is used to predict the most likely state or observation given a set of observations and the model's parameters. The advantage of using the Posteriori algorithm is that it is simple to implement and computationally efficient, making it well-suited for real-time prediction tasks. However, one disadvantage is that it can be sensitive to the initial conditions of the model, and may not always produce the most accurate predictions. In this script the Maximum a Posteriori (MAP) algorithm is used to predict the next observation in the sequence based on the previous observations. The MAP algorithm maximizes the posterior probability of the observation given the model's parameters. This algorithm is a variation of the Maximum Likelihood Estimation (MLE) algorithm, with the added constraint that the predicted observation must also be the most likely one given

the current state of the model. The formula for the MAP estimation is given as:

$$\hat{x} = \arg\max_x P(x|y) = \arg\max_x \frac{P(y|x)P(x)}{P(y)}$$

Here, *x* represents the predicted observation, *y* represents the previous observations, *P(x|y)* is the posterior probability of *x* given *y*, *P(y|x)* is the likelihood of y given *x*, *P(x)* is the prior probability of *x*, and *P(y)* is the marginallikelihood. In this script, the MAP algorithm is used to predict the next observation in the sequence based on the previous observations and the model's parameters. The advantage of using the MAP algorithm is that it incorporates prior knowledge about the distribution of the data, which can lead to more accurate predictions. Idea to use "Posteriori alghorithm" came from github project made by "Ajaypal91" (https://github.com/Ajaypal91/PredictingPrice-of-Cryptocurrency).

The graph Figure 1 represents a Gaussian Mixture Model (GMM) with 4 components. The nodes represent the different states of the model, while the edges represent the transition probabilities between these states. The edge labels show the weight of the transition, which is the probability of transitioning from one state to another. The overall structure of the graph represents the underlying probability distribution of the model, which is a combination of Gaussian distributions. The interpretation of this graph is that it gives us an understanding of the underlying structure of the data and how the different states are related to each other through the transition probabilities. The nodes represent the different states or components of the GMM, and the edges represent the transitions between these states. The edge labels represent the transition probabilities between the states, with values ranging from 0 to 1. These values indicate the probability of transitioningfrom one state to another, with a higher value indicating a higher probability. For example, the transition from state 0 to state 0 has aprobability of 0.694, while the transition from state 0 to state 1 has a probability of 0.277. This graph provides a visual representation of the GMM and the relationships between its states 1

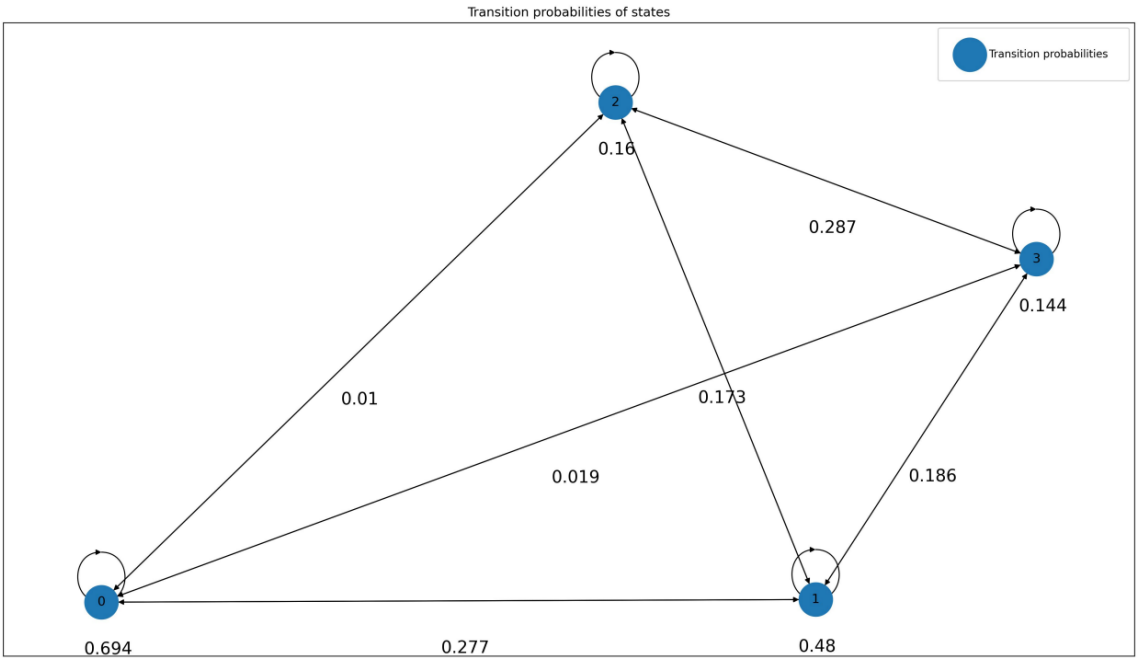

Figure 1. Visual representation of the Gaussian Mixture Model Hidden Markov Model Chain

In a Hidden Markov Model, the transition matrix plot Figure 2 displays the transition probabilities between various states (HMM). The starting state is represented by each row, and the ending state is represented by each column. The probabilities of changing from the starting state to the ending state are represented by the values in the matrix. The transition matrix in this particular instance has 4 rows and 4 columns, signifying 4 different HMM states (Fig. 3). The probabilities of changing between these states are represented by the values in the matrix. For instance, the probability of changing from state 0 to state 0 is

0.65302059, and the probability of changing from state 0 to state 1 is 0.1386918. These numbers indicate the likelihood that the system will switch from one state to another.

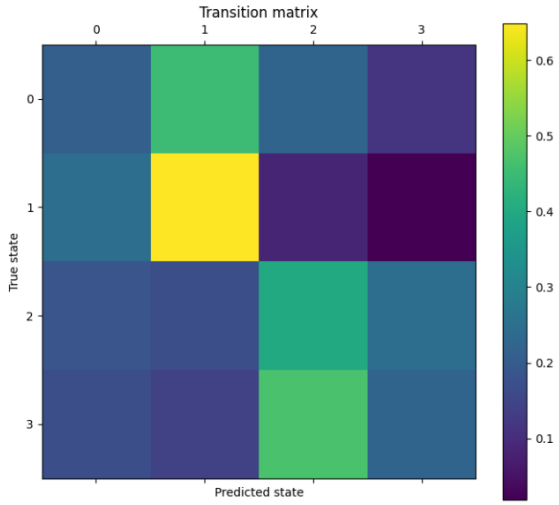

Figure 2. Transition matrix

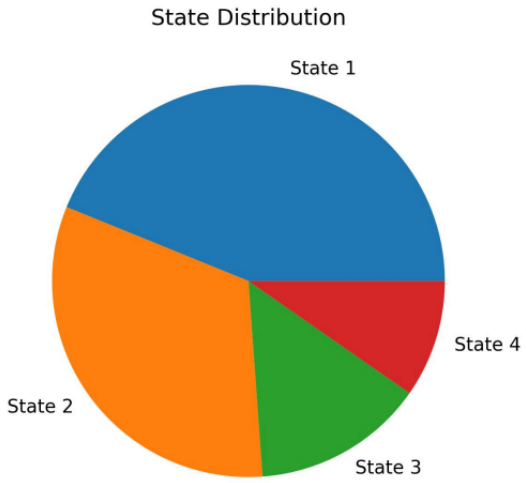

Figure 3. State distribution of the GMM-HMM model

5. CODE OVERVIEW

This Python-written program (Fig. 4) is used to forecast changes in a cryptocurrency's exchange rate. The predict function, which uses the trained model to predict the best values for the chosen features, is the primary focus of the code. A crucial part of the code is the predict function, which is in charge of forecasting future changes in exchange rates. The code begins by importing the necessary libraries and defining several functions. The fetch_data() function retrieves cryptocurrency exchange rate data from the internet using the requests library (Fig. 5). The save_data() function then saves this data in a CSV file. The read_data() function reads data from the CSV file and returns it in the form of a pandas DataFrame object. The extract_features() function processes the data using the pandas library and returns new features, which are then used to train the prediction model using the train_model() function. The train_model() function uses the GMM-HMM (Gaussian Mixture Model) approach and trains the model using the hmmlearn package. The predict function is used to predict the best values for the selected features based on the trained model. The function takes in two arguments, the trained model and the selected features. The function starts by defining the prediction range for the close-open/open, high-open/open, and open-low/open observations.

```
def predict(model, features):
    predFracChange = np
    .linspace(-0.1, 0.1, num=200)
    predFracHigh   = np
    .linspace(0, 0.1, num=20)
    predFracLow    = np
    .linspace(0, 0.1, num=20)
    observations = []
    for i in predFracChange:
        for j in predFracHigh:
            for k in predFracLow:
                observations
                .append([i, j, k])

    observations = np
    .array(observations)
    observationSeq = features[-10:]
    score = -9999999
    best = []
    for i in observations:
        seq = observationSeq
        s = model.score(seq)
        if(s >= score):
            best = i
            score = s
    return best, score
```

Figure 4. Program for forecast changes in a cryptocurrency's exchange rate

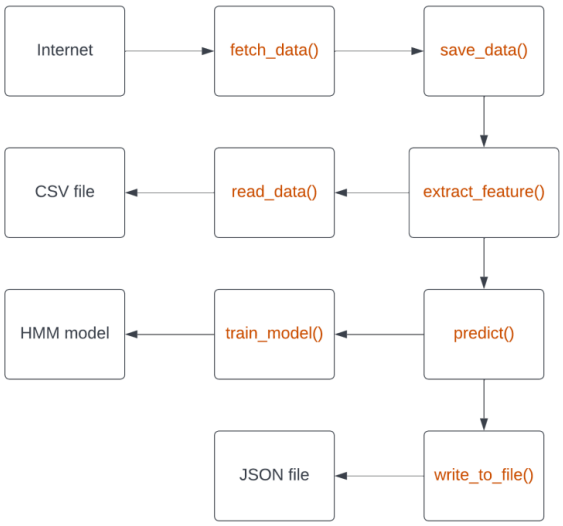

Figure 5. Block diagram of program

Then, iterating through the predetermined prediction ranges for each observation, it creates an empty list called observations. It adds the most recent observation values to the list of observations for each iteration. After the iteration is complete, the observations list is converted to an array and the last 10 observations from the features are stored in the observationSeq variable. The function then initializes a score variable with a very low value and an empty list called best. It then iterates over the observations array, and for each iteration, it calculates the score of the current observation sequence using the model.score() function. If the current score is greater than or equal to the previous score, the current observation values are stored in the best list, and the score is updated. Finally, the function returns the best observation values and the score. This value can be used to make predictions about future exchange rate fluctuations. The final part of the code involves calling the individual functions with the appropriate arguments and displaying the results on the plots using the matplotlib library.

In summary, this code uses the GMM-HMM approach and the hmmlearn package to predict future fluctuations in cryptocurrency exchange rates. The predict function plays a crucial role in this process, by using the trained model to predict the best values for the selected features. The code also includes several other functions to retrieve and process data, and to save and display the results.

6. RESULTS

The results of the GMM-HMM model trained on the cryptocurrency exchange rate data were analyzed and compared with the predictions made using the Prophet library.

Different numbers of states and mixtures were used to train the GMM-HMM model, and the prediction out comes were discovered to be consistent in all cases. The GMM-HMM model's predictions were found to be reasonably close to the actual exchange rate, but not quite accurate enough. All best predictions were found to be [0.1 ± 0.01, 0.1 ± 0.01, 0.1 ± 0.01]. This suggests that the predicted prices are rising with the same strength, as represented in Figure 6. The transition matrix Figure 2 plot also supports this, as all of the probabilities in the matrix are relatively similar, indicating that the model is not able to predict a clear direction for the cryptocurrency prices.

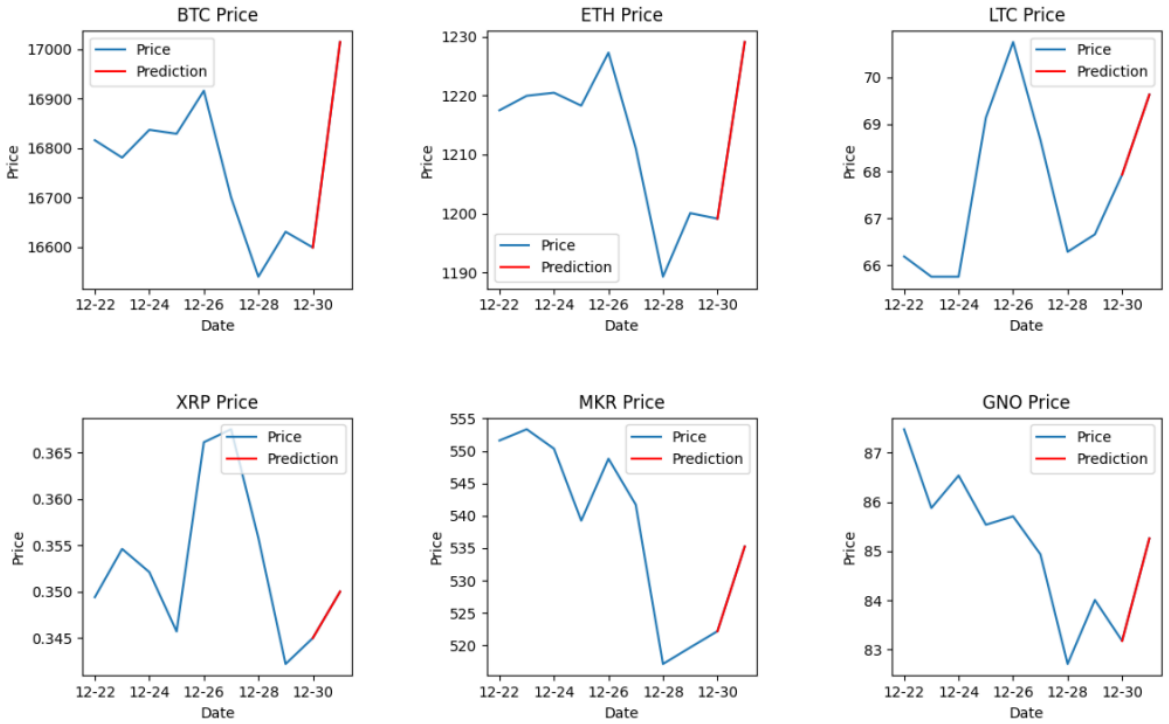

Figure 6. GMM-HMM model predictions (BTC/ETH/LTC/XRP/MKR/GNO-USD)

On the other hand, the predictions made using the Prophet library were found to be more accurate, with a smaller deviation from the actual exchange rate. The Prophet library uses a regression-based approach for time series forecasting Figure 7, which makes it more suitable for predicting the cryptocurrency exchange rate.

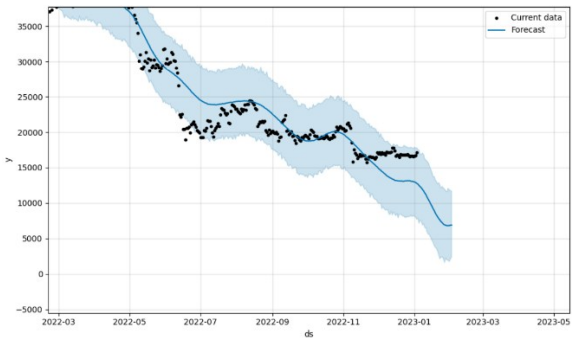

Figure 7. Historical data extended for prophet forecast (BTC-USD)

It is crucial to remember that forecasting the cryptocurrency market is a difficult task in and of itself, and the current strategy could still be improved. Adding extra features to the model, such as news sentiment analysis, may help to improve the predictions. It might also be advantageous to test various time series forecasting techniques to see if they outperform the current methodology.

7. CONCLUSION

This study used a Hidden Markov Model (HMM) approach based on a Gaussian Mixture Model (GMM) to forecast fluctuations in the exchange rate of a cryptocurrency. The Maximum a Posteriori (MAP) algorithm was used to train the model as part of the implementation of the GMMHMM approach using the hmmlearn library. However the GMM approach was found to produce similar or identical results across datasets, indicating that it is not a good option for forecasting cryptocurrency exchange rates. This study also demonstrated how challenging it is to forecast the cryptocurrency market and the potential for improvement by adding extra variables like news data and by investigating other time series forecasting techniques.

REFERENCES

[1] Raza S., Khan K., Guesmi K., Benkraiem R.: *Uncertainty in the financial regulation policy and the boom of cryptocurrencies*, Financ Res Lett. vol. 52(3), 2023, pp. 103515, DOI: 10.1016/j.frl.-2022.103515.

[2] Bommer W., Milevoj E., Rana S.: *The intention to use cryptocurrency: A meta-analysis of what we know*, Emerging Markets Review, vol. 55, 2022, pp. 100962, DOI: 10.1016/j.ememar.2022.-100962.

[3] Ozdamar M., Sensoy A., Akdeniz L.: *Retail vs institutional investor attention in the cryptocurrency market*, Journal of International Financial Markets, Institutions and Money, vol. 81(6), 2022, pp. 101674, DOI: 10.1016/j.intfin.2022.101674.

[4] Katsiampa P., Yarovaya L., Zięba D.: *High-frequency connectedness between bitcoin and other top-traded crypto assets during the covid-19 crisis*, Journal of International Financial Markets, Institutions and Money, vol. 79(3), 2022, pp. 101578,. DOI: 10.1016/J.INTFIN.2022.101578.

[5] Mahaleh M., Mirroshandel S.: *Harmony search path detection for vision based automated guided vehicle*, Rob Auton Syst, vol. 107, 2018, pp. 156–166, DOI: 10.1016/J.ROBOT.2018.06.008.

[6] Jaquart P., Dann D., Weinhardt C.: *Short-term bitcoin market prediction via machine learning*, Journal of Finance and Data Science, vol. 7, 2021. pp. 45–66, DOI:10.1016/j.jfds.2021.03.001.

[7] Ibrahim A., Kashef R., Corrigan L.: *Predicting market movement direction for bitcoin: A comparison of time series modeling methods*, Computers and Electrical Engineering. vol. 89, 2021, pp. 106905, DOI: 10.1016/j.compeleceng.2020.106905.

[8] Tian F., Gao Y., Yang C.: *Gmm based low-complexity adaptive machine-learning equalizers for optical fiber communication*, Opt Commun, vol. 517, 2022, pp. 128312, DOI: 10.1016/j.optcom.-2022.128312.

[9] Huque A., Ahmed K., Mukit M., Mostafa R.: *Hmm-based supervised machine learning framework for the detection of fecg r-r peak locations*, IRBM, vol. 40(3), 2019, pp. 157–166, DOI: 10.1016/j.irbm.-2019.04.004.

[10] Henrique B., Sobreiro V., Kimura H.: *Literature review: Machine learning techniques applied to financial market prediction*, Expert Syst Appl, vol. 124, 2019, pp. 226–251, DOI: 10.1016/j.eswa.-2019.01.012.

[11] Huang Y., Kou G., Peng Y.: *Nonlinear manifold learning for early warnings in financial markets*, Eur J Oper Res, vol. 258(2), 2017, pp. 692–702, DOI: 10.1016/j.ejor.2016.08.058.

[12] Mzoughi H., Benkraiem R., Guesmi K.: *The bitcoin market reaction to the launch of central bank digital currencies*, Res Int Bus Finance, vol. 63, 2022, pp. 101800, DOI: 10.1016/J.RIBAF.2022.101800.

[13] Yao K., Yu D., Deng L., Gong Y.: *A fast maximum likelihood nonlinear feature transformation method for gmm–hmm speaker adaptation*, Neurocomputing, vol. 128, 2014, pp. 145–152, DOI:10.1016/J.NEUCOM.2013.02.050.

[14] Jansen S.: Hands-On Machine Learning for Algorithmic Trading: Design and implement investment strategies based on smart algorithms that learn from data using Python, *Packt Publishing*, 2018.