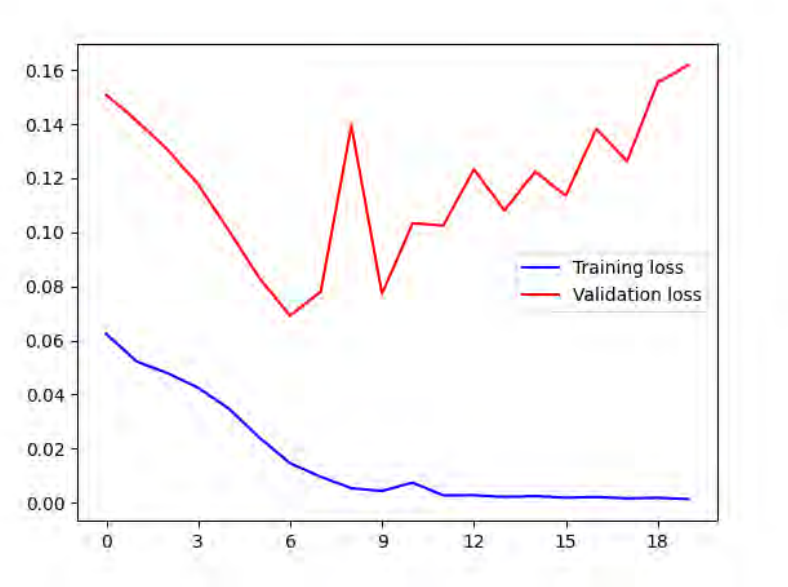
Modelling and Prediction of Prices for Products on Biofuel Markets

Thom Oosterhuis

August 24, 2021



Internship report

Modelling and Prediction of Prices for Products on Biofuel Markets

Thom Oosterhuis

August 24, 2021

VU, Amsterdam
Faculteit der Bètawetenschappen
De Boelelaan 1085
1081 HV Amsterdam

Host organization
Olyx B.V.
Herengracht 458
1017 CA Amsterdam

Preface

This report is written as part of the Master Project Business Analytics. It is the result of an internship at the company Olyx and describes the research of the author. The research problem that is investigated concerns the modelling and predicting of prices of biofuel products. Special thanks go to the supervisors of this project, which are Eliseo Ferrante (VU), Joppe Hülsenbeck and James Hitchcock (Olyx).

Abstract

The market for biofuels, consisting of both physical products and transferable emission rights, is an upcoming market as a result of increasing demand for cleaner ways of producing traffic fuels. Products on these markets are traded directly between buyer and seller, that might use the service of a brokerage firm to find a suitable counterparty. The prices of the products traded vary over time and depend on numerous factors. This report focuses on the modelling and predicting of these prices. Market prices are predicted using historic prices in the form of time series data. These historic prices include the prices of the product that is modelled. Second, the historic prices of similar products are used as input data. We build a one time period ahead predicting model, that predicts the price change relative to the last known price.
The main model is a Long Short Term Memory neural network that takes multiple fixed size time series as input. This LSTM model is capable of making predictions based on time series data that are more accurate than 'naïve' predictions that do not use any data. However, this is only the case for data where the correlation between time series is relatively high. Prices of individual transactions can be modelled based on market prices. Using a number of properties of a transaction, a regression model is able to model and predict the prices of transactions.

Contents

1 Introduction

In the market for biofuels, physical products and transferable emission rights are traded between seller and buyer, often using a intermediate party. The intermediate party has to make sure buyer and seller agree on the price for the product. Seller and buyer indicate an offer respectively a bid. The right price of a product that both parties can agree on, differs over time and depends on multiple external factors, including market conditions, historic prices, demand for fuel and regulations. Besides, every product has its own characteristics. Physical products for example have different quality and different chemical product specifications.
Olyx acts as the intermediate party on the market for biofuels. Brokers at the company intermediate trades in different physical products such as biodiesel as well as nonphysical products in the form of emission rights. The use of - and therefore trade in - biofuels in the European Union has grown over the last ten years as a result of regulations. Suppliers of fuel for road transportation have the obligation to reduce the CO_2 emissions that are released when producing the fuels, the upstream emissions. They mainly do so by blending their fossil fuels with biofuels, or by buying emission rights from other supplier that did so. This research focuses on the question of how to model and predict the prices of non standardized products in the biofuel market. We use historic price data and human made price assessments to answer this question. The main research question of this report is

> To what extent can we model and predict prices of different biofuel products, using historic price data and human made price assessments?

We can break down this question using some smaller subquestions. First, we would like to investigate whether we can predict prices using time series data on historic prices only. This leads us to our first subquestion

- Can we predict biofuel market prices from historic price data in the form of time series only, using machine learning prediction models developed for time series?

Furthermore, there are two different kinds of prices we would like to model and predict to fully answer the research question. First, there is the general market price, the price that changes over the day. This price can be interpreted as a reference price that will be the base for the determination of the price of a particular deal. The price for a particular deal however does depend on other factors as well. These include product characteristics as mentioned above, as well as for example the parties involved in the deal or the product volume of the deal. This means that in order to fully answer the main question, our second subquestion focuses on the modelling of the prices as traded on the market

- Can we predict the price of a product as it is actually traded between two parties as a function of 1.) the market price and 2a.) chemical properties of the product and 2b.) the product volume of the trade?

A model that can predict and describe dependencies of different prices in the biofuel market will help participants in the market to explain market trends.

This in turn will make new opportunities visible and will give an incentive to brokers to spot potential trade possibilities. Secondly, a model for the price of a specific product will give insight in the dependency between price assessments on one side and the actual prices, the properties of a deal and the properties of a trade on the other side. In doing so it can help to identify as well as confirm business logic on what properties of a product play an important role in determining its price.
The report starts with an overview of literature on similar research and different modelling techniques. Some different models and methods are investigated and explained in more detail. Next, we describe the data that is available on biofuel prices. We propose to use a few models which we believe might be suitable for our research. After looking into some details of how these models can suit our purposes, we create an experimental setup. We evaluate the performance of different models and investigate how we can improve these models by tuning their hyperparameters. For each model, we identify what data seems to add to the explanatory power of the model. Finally, results are evaluated and compared, leading to some conclusions and recommendations for the business case, as well as for further research.

2 Literature

In the following section, we have a look at different models and techniques that we will use throughout this report. We start with a discussion on similar research and different methods and techniques used. Most models selected will be able to work with time series, as this suits most of our data available. First we discuss the use of regression models. Second, we discuss the use of AutoRegressive Integrated Moving Average models. These models are a good start when analyzing time series. Third, we discuss neural networks and try to give some insight in how these work. Due to the relative complexity of these models, their exact working might not always be immediately clear. By discussing their most important principles we try to reveal their basic mechanisms. A neural network can take many types of data as its input. A type of neural network that is known to work well with time series data, is a Recurrent Neural Network. An instance of this is a Long Term Short Term neural network. We will discuss in more detail how both these networks work and why they suit time series data. Finally, we discuss different error metrics that can be used to compare the models. Most of these models will be trained on certain part of the data. The evaluation of the model then will use a different part of the data. We shortly discuss this splitting of the data.

2.1 Similar works

The modelling and forecasting of time series data is an important subject in many applications, for example in economic or finance applications. AutoRegressive Integrated Moving Average models offer a relative simple way to fit time series. This allows for better understanding of the data and for making predictions on future values of the time series. More advanced methods using machine learning algorithms and deep learning also exist. With advancement in computational power of computers and development of these methods, such methods are now suitable for modelling and predicting time series data [9]. Namini shows that Long Short Term Memory Neural Networks are suitable for modelling time series data and give superior predictions when compared to ARIMA models [9]. To study the relationships between two time series data sets, canonical correlations are a popular method. This method can model simple linear correlations between time series. A nearest neighbor overlap method as introduced by Roy improves on these methods such that it also allows for modelling more complex non linear relations between time series [12]. Huang and Sun focus on the prediction of prices in the Hubei carbon trading market [14]. They decompose the time series data using empirical mode decomposition. Using the Partial Auto Correlation Function, they determine the relevant input variables of the model. They continue to train a neural network using back propagation. This neural network predicts decomposed subsequences. From these sequences, the actual predicted values for the carbon prices are calculated.
Park and Bae in their paper on using machine learning algorithms for making predictions on housing prices, have a setup that is comparable to ours [11]. They use different machine learning algorithms to predict housing prices. Given the listing price of a house, their model predicts whether the actual closing price is greater than or less than the listing price. Our subquestion on the closing prices of individual deals, is similar in setup. In our case, we have a general market

price and we would like to predict the actual closing price of a deal. Similar to the prediction of these housing prices, we use features of a deal, together with market data, to make a prediction for the price of this deal. Park and Bae compare four different algorithms and find that the Repeated Incremental Pruning to Produce Error Reduction (RIPPER) algorithm out of these performs best.

2.2 Regression

Regression is a model technique that is simple and widely used. Partly because of its simplicity, it produces models that are easy to interpret and understand. At the same time, it can produce models that make relative accurate predictions. Since these models are widely known, we discuss them only shortly. A regression model predicts the value of a continuous target variable using a vector of input variables. In a simple case this target is a linear function of the inputs, but other functions can be used as well. The output of a linear regression model is calculated using the input variables

$$\hat{y}(x) = \sum_{i=0}^{I} \theta_i x_i.$$

Where $x_0 = 1$. Given a sample (X, Y), the parameters $\Theta = (\theta_0, \theta_1, \ldots, \theta_I)$ in this equation can be estimated using the Ordinary Least Squares method. In this method, the sum of squares of the differences between the observed and estimated dependent variable

$$\sum_{i=1}^{n} (y_i - \hat{y}_i(x))^2$$

is minimized over Θ.

2.3 Time series and ARIMA models

A time series $(X)_t = (X_0, X_1, \ldots, X_t)$ is defined as a series of discrete data points. Each data point X_i is a repeated observation of an event at time t. For convenience, we choose the time distance between t and $t+1$ to be constant for all $t \in \mathbb{Z}$. An important concept when modelling time series is stationary. A time series is weakly stationary if for any k, $\mathbb{E}X_t$ and $\mathbb{E}X_tX_{t+k}$ exists and are independent of t. In practice a time series may often not be stationary. However, taking differences $\bigtriangledown X_t = X_t - X_{t-1}$ may generate a stationary time series from a time series that is not stationary itself. The difference operator can be repeated d times and can in this way reduce a polynomial trend of degree d to a constant.
A Moving Average process specifies an output variable as a linear combination on current and past values of a stochastic term. For a normal distributed variable $\epsilon \sim \mathcal{N}(0, \sigma^2)$ we denote an MA process of order q as $X_t = \mu + \epsilon_t + \sum_i^q \theta_i \epsilon_{t-i}$, with μ a constant. The value of X_t in an Auto Regressive process depends linearly on its previous values and is denoted as $X_t = c + \sum_i^p \phi_i X_{t-i} + \epsilon_t$, with c a constant.
An ARIMA(p, d, q) model can be fitted to a time series to get an idea how the time series is constructed, as well as to make predictions. The I stands

for integration and refers to taking differences of the original time series. This is done d times. The autoregressive and the moving average part then model the integrated time series as a process that contains an auto regressive part of degree p and a moving average of degree q.
A method to apply ARIMA models to a time series to find a good fit for past values of the time series is the Box-Jenkins method [2]. This method divides the finding of a model in three phases. The first phase concerns the identification. Using a unit root test, we can determine whether the time series is stationary. If not, the time series can be differenced to see if the new series does produce a stationary series. This gives the value for d. Next, the parameters p and q need to be determined. The Auto Correlation Function and the Partial ACF can help with this. The former function denotes the correlation between X_t and X_{t-h}. The latter does the same, but adjusted for the intervening observations $X_{t-h+1}, \ldots, X_{t-1}$.
In the second phase, the model parameters are estimated such that they minimize a specified loss. This is done using numerical methods. The model with estimated parameters is checked in the third phase. The residuals of the data points after applying the model are checked. By assumption of the model they should follow a Gaussian distribution.
To check if a time series is indeed stationary, we use the Dickey-Fuller test. This test is developed by Dickey and Fuller (1979) and can be used to test for stationary of a time series [4]. The test has the null hypothesis that a unit root is present in the time series. When this is the case, the time series is not stationary. The alternative hypothesis of the test is that the time series is stationary.

2.4 Neural Networks

Artificial Neural Networks have a basic structure of small processing units. These units are connected with weights. Nodes are connected in different ways. A distinction we make is between networks of nodes that contain cycles, also called Recurrent Neural Networks, and neural networks that do not contain such cycles. We will start studying the latter in more detail, since these are more easily described. A wide used form of these networks is the MultiLayer Perceptron network. MLP networks consist of different layers. They have an input layer, one or multiple hidden layers and an output layer. Figure (2) gives a schematic representation of such an network. There is a collection of nodes with initial value based on the data, the input layer. A hidden layer consists of a different collection of nodes. Each node has a value that is calculated based on a (linear) combination of the nodes from the previous layers, for example the input layer or another hidden layer. The final layer is called the output layer. The values of this layer can be seen as the output values. Put simply, we could see the network as a function $\hat{y} = f(x)$, where f represents the network, x the input and $\hat{y}$ the output. MLP can been seen as universal function approximators [1]. In practice we often have data (x, y) and we would like to find a network f that maps x to y, in such a way a certain error $\epsilon = y - f(x) = y - \hat{y}$ is minimized. More general, this error can be any loss function $L_{(x,y)}(f)$. In order to minimize the loss, we can take the partial derivatives over the different network weights with respect to the loss function. To be able to do so, first we will take a closer look at how the MLP network maps the input to the output.

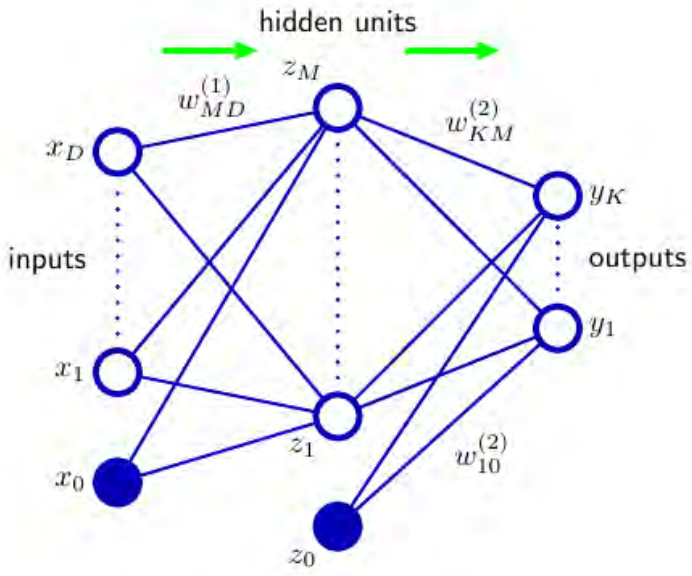


Figure 2: Display of a neural network, by Bishop [1].

2.4.1 Forward propagation

The values for the units in the input layer of the network are calculated using the data X and their associated weights. For a vector $X = (x_1, \ldots, x_I)$ we could for example choose an input layer with I nodes. The i^{th} node could then take x_i as input value. For the first hidden layer h_1 the input value of node j will be

$$a^j_{h_1} = \sum_{i=1}^{I} w_{i,j,1} x_i + w_{0,j}. \tag{1}$$

The value of the corresponding node h^j will be calculated by applying an activation function to this value

$$b^j_h = \theta_h(a^j_h). \tag{2}$$

The activation function is often chosen differentiable and non linear. This choice depends on the kind of model. Common choices for $\theta(x)$ involve the logistic sigmoid function or the hyperbolic tangent function. The values of the nodes in hidden layer l are calculated similarly using the weights $w_{i,j,l}$ and the values of the nodes in hidden layer h_{l-1}

$$a^j_{h_l} = \sum_{i=1}^{|H_{l-1}|} w_{i,j,l} b^i_{h_l} + w_{0,j,l}, \tag{3}$$

$$b^j_{h_l} = \theta_{h_l}(a^j_{h_l}). \tag{4}$$

Where

- h_l^j denotes node j in hidden layer l .
- $b_{h_l}^j$ (or alternatively $z_j^{h_l}$) denotes the value of node j in hidden layer h_l.
- $H = \{h_l^1, h_l^2, \ldots, h_l^j, \ldots, h_l^l\}$ denotes the set of nodes in hidden layer h_l.
- $w_{i,j,l}$ denotes the weight from node i in layer $l-1$ to node j in layer l.
- $w_{0,j,l}$ denotes the bias of node j in layer h_l.
- $a_{h_l}^j$ (or a_l^j for ease of notation) denotes the activation value of node j in hidden layer l.

2.4.2 Output layer

The calculation of the values of the nodes in the output layer is similar to the calculation of the values of the nodes in a hidden layer. The activation function used in this calculation depends on the particular problem. For regression problems, the activation function in the output layer is often chosen to be the identity function. For multiple binary classification problems, the final activation is often the logistic sigmoid function.
Based on the output values in the output layer a loss function can be chosen to evaluate the outputs. A loss function used for classification could for example be defined as

$$\mathcal{L}(x, z) = -\log p(z|x),$$

where x denotes the actual category and z denotes the category assigned by the neural network. Other loss functions include the Mean Square Error, that is suitable for continuous output variables. These error metrics will be discussed in more detail below.

2.4.3 Training the network: BPTT

Given the outputs of a feed-forward neural network as described above, we would like to evaluate the gradient of the error function of the outputs $E(w)$. Backward Propagation Through Time is a widely used algorithm to train a neural network as described above. The algorithm looks at the gradient of the loss function with respect to the model parameters, the weights. We can train the network in two stages. In the first stage, the partial derivatives of the error function with respect to the weights are evaluated. In the second stage, the weights are iteratively updated, using for example gradient descent. These procedures can be repeated iteratively. Assuming a non convex error $E(w)$, this will find a local minimum of the error function.
We take the derivative of the loss function with respect to the weights. Note that the error depends on the weight $w_{i,j,l}$ through the summed input a_j [1]. Using the chainrule we have

$$\frac{\partial E}{\partial w_{i,j}} = \frac{\partial E}{\partial a_j} \frac{\partial a_j}{\partial w_{i,j}}. \tag{5}$$

Define

$$\delta_j = \frac{\partial E_n}{\partial a_j}. \tag{6}$$

Using the equation $a_j = \sum_{i=1} w_{j,i} z_i$ we can write

$$\frac{\partial a_j}{\partial w_{ji}} = z_i. \tag{7}$$

Now we substitute equation (7) and (6) into equation (5) to obtain

$$\frac{\partial E_n}{\partial w_{ji}} = \delta_j z_i. \tag{8}$$

For the δ's in the hidden units we have

$$\delta_j^{h_l} = \frac{\partial E_n}{\partial a_j} = \sum_k \frac{\partial E_n}{\partial a_k} \frac{\partial a_k}{\partial a_j}. \tag{9}$$

The k then runs over all units to which node j sends a connection.
Using (6) and definitions (3) and (4), we get

$$\delta_j = \theta'(a_j) \sum_k w_{kj} \delta_k, \tag{10}$$

the backpropagation formula.
To summarize, after calculating the values of the nodes using the data and the feedforward procedure, we can calculate the values of the δ_k's in the output layer. After that we can calculate the δ_j for the hidden units. Finally, using equation (8) we can evaluate the derivatives of the error function.

2.4.4 Gradient descent

Most training algorithms minimize an error function in an iterative procedure. Weights are adjusted in a sequence of steps. For each step, we distinguish two stages. The first stage involves evaluating the derivative of the error function. This stage called back propagation is discussed above. In the second stage, these derivatives are used to make adjustments to the weights. These adjustments can be made in different ways. One of the simplest techniques to do this is gradient descent and is described by Rumelhart [13]. Weights are updated such that they move a small step in the direction of the negative gradient

$$w^{\tau+1} = w^{\tau} - \eta \bigtriangledown E(w^{\tau}).$$

Here $\eta > 0$ is called the learning rate. The error set is defined with respect to the training set, so at each step the error and its gradient need to be evaluated on the entire training set. Algorithms can use the whole training set, or split the training data in different batches. The size of these batches is the batch size. We will see later that both the learning rate and the batch size are important hyperparameters when tuning a neural network. Besides gradient descent other optimization schemes to update the weights exist as well. Many of them are more powerful than simple gradient descent [1].

2.5 RNN & LSTM

The general neural network as described above does not take into account the past. This means that when the network is trained on data $X_t = (x_{t-l}, \ldots, x_t)$, it does not take into account values $(x_0, \ldots, x_{t-l-1})$. In order to solve this, Recurrent Neural Networks have extra connections on top of the connections between nodes of different layers as we saw above for the MLP network. These extra connections are between nodes within the same layer, such that the nodes in the network at time t connect to the nodes of the network for other t as well.

LSTM

Long Short Term Memory networks are a special case of RNN, introduced by Hochreiter [7]. This kind of network has been observed to perform well on time series data, since it is able to learn long term dependencies. A clear explanation of this neural network is given by Olah [10]. A graphical display of a node in an LSTM neural network is shown in Figure (3), which is obtained from Olah. The

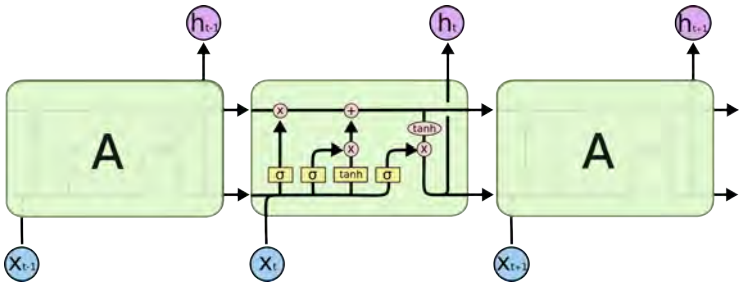


Figure 3: Display of LSTM node, by Olah [10].

main characteristic of an LSTM network is that nodes (or cells) in the network have a cell state C_t for each cell at time t, next to the cell output value h_t. Given a previous state C_{t-1}, a data input x_t and a cell output h_{t-1}, the new cell state and output are calculated using the weights W and the bias b as follows.
A forget gate is calculated

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f).$$

A second sigma layer is the input gate layer, that is calculated

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i).$$

The new candidate state cell is calculated

$$\tilde{C}_t = \sigma(W_C[h_{t-1}, x_t] + b_C).$$

The above are used to calculate the new cell state

$$C_t = f_t C_{t-1} + i\tilde{C}_t.$$

At this point the terminology also makes sense. The interpretation for the new cell state is that it depends on the previous cell state, but part of that is forgotten, as determined by f_t. Next to that, the cell state depends on new

input, that is a function of the previous cell output and the data input. This new cell input is then filtered using i_t, that determines what part of the new cell state is actually kept. This together produces the cell state to be used for the cell at time t, C_t.
The final output of a cell h_t depends on the cell state C_t and the cell output o_t that is also used in the MLP neural network

$$\begin{aligned} o_t &= \sigma(W_o[h_{t-1}, x_t] + b_0), \\ h_t &= o_t \tanh(C_t). \end{aligned}$$

2.6 GARCH

In financial price data on for example stock returns, it is often observed that the time series have conditional variances that are not constant [15]. This can be interpreted as 'turbulent times' in the market, where a day with big changes in price is usually followed with another such day. A model that can account for this property is the Generalized Auto Regressive Conditional Heteroscedasticity process. A time series is GARCH(1,1) if for non negative α, θ, ϕ and a i.i.d. sequence Z_t with $\mathbb{E}[Z_t] = 0, \mathrm{Var}[Z_t] = 1$, it satisfies the following system of equations

$$\begin{aligned} X_t &= \sigma_t Z_t, \\ \sigma_t^2 &= \alpha + \phi\sigma_{t-1}^2 + \theta X_{t-1}^2. \end{aligned}$$

In this application, the variable σ_t has the interpretation of volatility of the time series X_t at t. We usually observe time series $X_1, \ldots, X_t$. We then assume $Z_t \backsim \mathcal{N}(0,1)$. The parameters α, ϕ, θ we can estimate using maximum likelihood. Conditional on information at t we have $X_t = \sigma_t z_t \backsim \mathcal{N}(0, \sigma_t^2)$. For the joint distribution we have

$$\begin{aligned} f(X_0, \ldots, X_T; \alpha, \phi, \theta) &= f(X_0; \alpha, \phi, \theta) f(X_1, \ldots, X_T; \alpha, \phi, \theta) \\ &= f(X_0; \alpha, \phi, \theta) \prod_{t=1}^{T} f(X_t | X_{t-1}, \ldots, X_0; \alpha, \phi, \theta) \\ &= f(X_0; \alpha, \phi, \theta) \prod_{t=1}^{T} f(X_t | X_{t-1}; \alpha, \phi, \theta) \\ &= f(X_0; \alpha, \phi, \theta) \prod_{t=1}^{T} \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp - \frac{X_t^2}{2\sigma_t^2}. \end{aligned}$$

From this, we drop $f(X_0; \alpha, \phi, \theta)$ and take logs, to obtain the conditional log-likelihood function

$$L(\alpha, \phi, \theta) = \sum_{t=1}^{T} \big[\log(2\pi) \ \log(\sigma_t^2) - \frac{X_t^2}{\sigma_t^2} \big].$$

Using a numerical approach, we minimize the negative log likelihood function over α, ϕ, θ to obtain an estimation for these parameters.

2.7 Error metrics

To evaluate the performance of different models, we look at different error measures as evaluated on the predictions made by the model. A widely used metric is the Mean Squared Error. It compares every prediction against its actual value, the difference between these two is the error. These errors are squared and their mean is calculated

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

For models that make predictions of one step ahead time series, the MSE only might only have meaning when it is compared to the MSE of other models. Therefore we may create a base model, i.e. a model that makes predictions without any data. In the case of one step ahead time series, a good baseline model is the 'zero difference model'. This model predicts $\hat{Y}_{i+1} = Y_i$. Or when predicting the differences, it predicts $\bigtriangledown\hat{Y}_{i+1} = 0$.
A similar error metric that makes this same comparison but expresses it in a single number is the Mean Absolute Scaled Error, (MASE). This error metric has the advantage that it is independent of the scale of the data. It was proposed by Hyndman [8] and is defined as

$$\text{MASE} = \frac{\frac{1}{J}\sum_j \mid e_j \mid}{\frac{1}{T-1}\sum_{t=2}^{T} \mid Y_t - Y_{t-1} \mid}.$$

Where $e_j = \hat{y}_j - y_j$ denotes the error at j. If the value of the MASE is greater than one, the predictions are worse than the average one-step naïve forecast. Conversely, when the MASE has value less than one, it shows the model actually has explanatory power and could be preferred over the baseline model.

2.8 Train/test split of data

In most cases, we create a model to make predictions about future prices. We would like to evaluate to what extend the model is able to produce predictions that are close to actual prices. Therefore we create a setup where we use part of the data to evaluate the model exclusively. This means we do not use this part of the data to fit the model parameters. This part of the data is called the test data. The data that is used to train the model and estimate the parameters on the other hand, is called the train data. The latter part of the data usually comprises about 70-90% of the data set. When evaluating the model using the test data, the dependent variables are given as an input to the trained model. The model will then produce predictions $\hat{Y} = f(X)$, where Y denotes the dependent variable, X the independent variable and f the model. The predictions $\hat{Y}$ can be compared to Y, since these data is available to us. Using some error metric, such as the MSE or the MASE as described above, predictions and actual values can be compared. This gives some indication of how well the model is able to perform. When the model is able to make accurate predictions on the test set, this might be an indication that the model will be able to make accurate predictions on future data, for which we do not know the actuals Y.
In the case of time series data, the split of the data into train and test data is a certain date. The train data consists of all the data before this date, while

the test data consists of all data after this date. In this way, the setup where the test data is available only after the model is trained, is comparable to the situation where more data becomes available as time passes.

3 Research Method and Data Gathering

In the two following chapters we discuss the application of different models that were introduced in the previous chapter. We start with a more detailed description of the available data. We will see different dependencies between time series exist. Certain prices of products are highly related since the products are very similar. We look into this, as well as different ways to take this into account in our models. For the model section we look at how we can apply the relative simple ARIMA model to our data.
Second, the LSTM models are studied. We look how we can apply them to different data sets and in different settings. To validate the use of LSTM models to our data, we create an experimental setup where we test whether the model setup allows to learn a trivial dependency in a time series.
Next we investigate the use of regression models to answer the second part of the research question concerning the prediction of the prices of a particular deal.

3.1 Form of research

As time series are a natural way of representing the prices over time, we choose models that in general work well with this kind of data. First we use ARIMA models to get familiar with the data. These models are widely used for time series data. They are relatively simple and give us insight in patterns that exist in the data. Second we look at models in which the price of a product is modelled and predicted using only the data of the historic price of the product itself. We use LSTM models for this, as they are known to work well on time series [1]. For the modelling and predicting of the price of a particular deal we use regression. As an input for this regression model we use the market price and certain properties of the deal. Regression models are relatively simple and therefore easy to interpret. Despite this, they can be very powerful in the modelling of linear relationships. As we will see later, restricting ourself to this kind of relationship will come at limited loss only, while a big part of the dependencies can indeed be modelled in linear way.

3.2 Data description

We use different kinds of data. Part of these data concerns the prices of products on the biofuel market directly. Other data is expected to be related to these prices. Some of these data on prices comes from Olyx internal information sources, such as the deals intermediated by the company. Other data are created by brokers at Olyx. Finally, we consider relevant external data. These data comes form third parties and is identified as potentially relevant by experts in the field.

Internal data

Olyx is a brokerage firm that intermediates trades between sellers and buyers on the biofuel market. This makes that from these transactions, the deals, Olyx has the information on concrete trades in the biofuel market. Secondly, many communication and negotiating usually takes place before closing a deal. Therefore, brokers know about market trends and developments. This gives

them some sense of developments and trends in the biofuel market. They use this information to give an indication of current market prices. We first discuss in more detail on the deal data. Secondly, we look at the data on the price assessments made by the brokers.

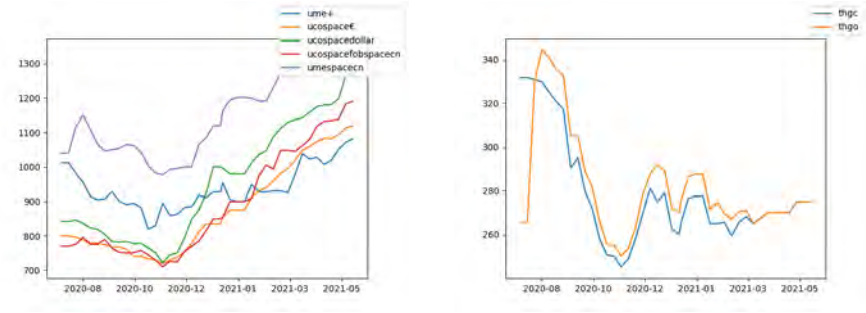
Deals

When a deal is closed by a broker at Olyx, some concrete information about what is happening in the biofuel market is revealed. The deal data gives information about a product, its specifications, the price of the product and the buyer and seller. Given that buyer and seller agree on a price for their deal, this means that presumably both parties think the price is reasonably at the given moment. They do not expect to get a better price somewhere else. As Olyx grows bigger, the part of the total transactions on the biofuel market they intermediate grows. Especially for frequently traded standardized products, the deal data may reveal much information about the right price of a product at a given moment.

Price updates

Besides the prices of the products as traded between two parties in a deal, there are the market prices. These prices give an indication of current market conditions. At Olyx, brokers are continuously in contact with market participants. This way, they hear about the latest market developments, the products that are offered and the products that are in demand. This information is valuable for the work of the brokers, in which they have to match supply and demand of products in the biofuel market. The real value of this information comes when the information is shared between brokers within the company. Brokers regularly share information with each other, either ad hoc or in daily meetings. This combined market information makes that a broker can say with some certainty what the 'right' price of a product is at the moment. This estimation of the market price, is called the price assessment. The interpretation of this assessment is that if a trade would be closed, the price of the product would be close to the price assessment. In practice, brokers often have a offer and a bid from respectively seller and buyer. To close a deal between these two, seller and buyer have to move to a price that both parties agree on. The price assessment of the broker will then usually be a price between this bid and offer. Based on information and experience, the broker can decide the price should be closer to the offer or the bid price.
For frequently traded products, brokers will give price assessments a few times a day. Whenever they have new information, for example a new offer or a new bid, they will update the price assessment of the concerning product. The assessment of the market prices does not involve precise calculations, but rather is based on experience to make a judgement on the right price. Moreover, there is a subjective part to the judgement. Price updates therefore can lead to differences in opinion between brokers. This is encouraged within the company, since discussion might reveal market mechanisms and in such lead to better price assessments.
These price assessments are important data in our research, as this is often as close as possible to what one could call 'the price of a biofuel product'. Therefore, we will regard the price assessment as the market price of a product.

This means that when we would like to predict the future price of a bio fuel product, in fact we are predicting how the broker will assess the product price in the future.
An example of some of these price assessments is shown in Figure (4), that shows the daily averages of the price assessments of different variants of UCO and UCOME in Figure (4a) and of price assessments of different variants of THG ticket prices in Figure (4).



(a) Prices for different variants of UCO UCOME.

(b) Prices for THG Conventional and THG Other.

Figure 4: Price Assessment Daily Averages for business days.

Exchange rate

Since different prices are listed in different currencies, these prices are converted such that they are displayed in the same currencies. For this, we use the daily exchanges rates.

Oil price

Contrary to most bio fuels, oil is more frequently traded. Therefore, different financial exchanges offer futures based on the oil price. This makes that there is clear data on the oil prices and that it is widely available.

3.3 Application of selected techniques

ARIMA

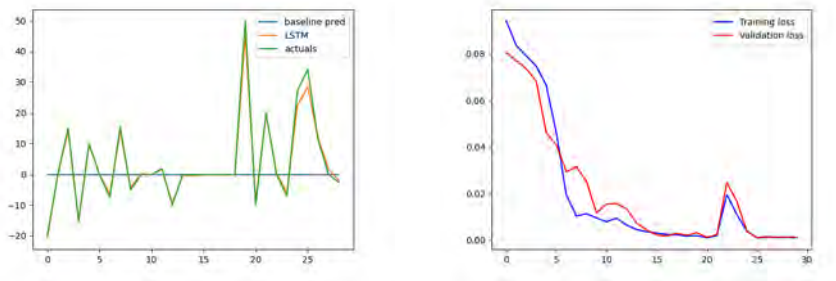
We use ARIMA models to model time series data to better understand the data or make predictions on the future of the time series. ARMA models assume stationarity of the data, so trends and seasonalities in time series data are removed. This can be done by taking differences of the series, this part of the process forming the Integrated part of the model. As our data for a large part comes into time series, or can be easily represented a such, the ARIMA model is a good first choice. We will use the Box-Jenkins approach for estimating the ARIMA models, as described in the previous chapter.

LSTM

Since we would like to investigate whether prices can be predicted using historic data, the input in our LSTM model setup consist of the historic price differences $(x_{t-1}, x_{t-2}, \ldots, x_{t-k})$. In our data, the time between two prices is not constant. In addition, some products have more price updates per time unit than others. As our models do require a constant interval between prices, we take daily or weekly averages of prices. These averages then form the sequence $X_t = (x_t, x_{t-1}, \ldots)$. On these series we can apply the difference operator. Also, as described in the section on ARIMA models, we can take differences multiple times. Due to the availability of data we take weekly averages of prices, unless specified else. From these prices, we take the weekly differences $\bigtriangledown x_t = x_t - x_{t-1}$. To make the data independent of the scale, we rescale the data into the interval $[-1, 1]$.

LSTM Model validation

In a perfect case, the input data contains all the information that is needed to make a one step ahead prediction $\hat{X}_t$ that is equal to X_t. To test our model setup we would like to test whether the model in this case is indeed able to learn from these data. For this purpose, we shift the input data by one time step ahead, such that the input data becomes $X_t = (x_{t-29}, x_{t-28}, ..., x_t)$ for predicting $Y_t = y_t = x_t$. So the model should learn to act as the identity function on the final value x_t of the vector X_t, ignoring the other 29 values in X_t. The predictions this model makes are shown if Figure (5a). The predictions made by the LSTM model are close to the actual values. The loss of the predictions made by the model after each iteration through the network is shown in Figure (5b). We see the model learns to make predictions with a lower loss as the number of iterations through the network increases. The hyperparameters used for this experiment as well as the MSE of the predictions on the validation set are listed in Table (1). For reference the MSE of the baseline model on the same validation set is also listed for comparison. It turns out the model can indeed learn to make accurate predictions.



(a) Predictions of daily price differences of oil price by perfect model.

(b) Loss per iterator over the network.

Figure 5: Validation experiment 0.

Table 1: Hyperparameters for experiment 0.

	Value
Sequence length	30
Epochs	30
Learning rate	0.001
Batch size	1
Train/val split fraction	0.700
Size training set	135
# nodes in LSTM layer	16
MSE on val. set	3.021
MSE of baseline model on val. set	221.957
Features	diff_ume+
p	1
n	193

3.4 Regression

Regression models are used for estimating the relationships between a dependent variable and independent variables. In our setup, we would like to predict the unit price of a deal, which is our dependent variable. We will use the unit price to make sure the dependent variable is expressed on a single scale. We notate this variable of our data by $Y = (y_1, \ldots, y_n)$. The first independent variable we use is the market price as assessed by the brokers. Since this price varies over time, we take for each deal the weekly average price assessment of that week. This means we have data points $(X, Y) = ((y_1, x_1), \ldots, (y_n, x_n))$ where x_i is the price assessment of the corresponding week in which the deal with unit price y_i is closed. Further independent variables we include concern other properties of the deal. These include the volume of the deal, as possibly the unit price might be influenced by the total volume of the deal. An other important factor of the price of UCO as identified by brokers is the FFA, a chemical property of the feedstock that is of importance for manufactures producing diesel out of UCO. By rule of thumb, a lower value of FFA mean usually a higher price of the UCO. In our regression model we will investigate whether this dependency can also be seen from the data.

4 Data Analysis

In the previous chapter we introduced the different datasets on biofuel prices and related data. In this chapter we give a description and a first analysis of these data. We classify different products in the biofuel market and look how these might relate to each other.

4.1 Product dependencies

The different biofuel products that are priced at Olyx are often related to each other. This also shows in the pricing of the different products, where a price change in one product will usually go hand in hand with a price change in an other product. This connection might be one of causality. When one looks at the product chain, this makes sense from an economic point of view. For some groups of products, these correlations between prices of products are more clear than for others. We illustrate with some examples.

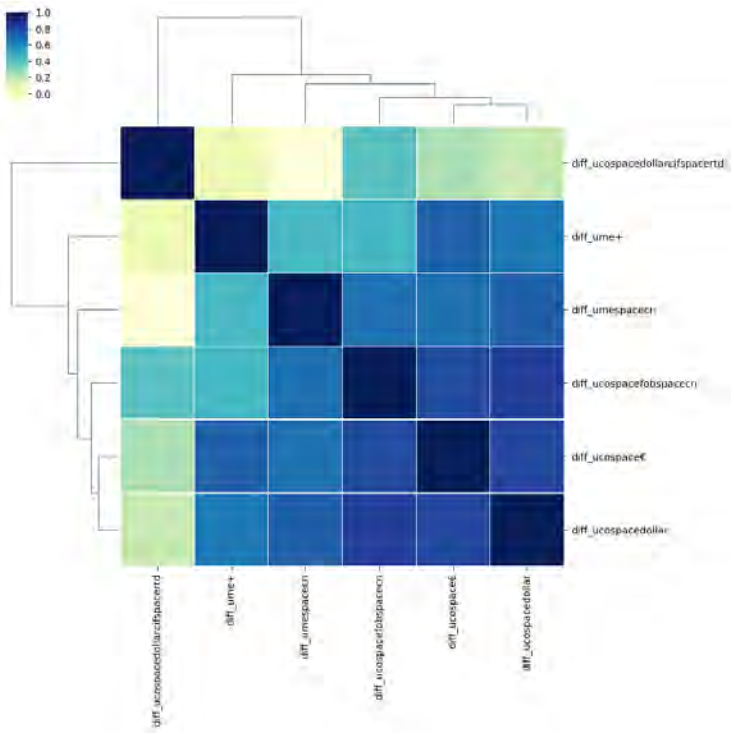


Figure 6: Clustermap of UCO and UCOME prices.

4.1.1 UCO and UCOME

UCOME is a biodiesel produced using the raw material UCO, Used Cooking Oil. Refineries import the raw material often from China and other countries in South East Asia, where UCO is produced as a rest or waste product from factories. The raw material is then used to produce biodiesel. This is mostly

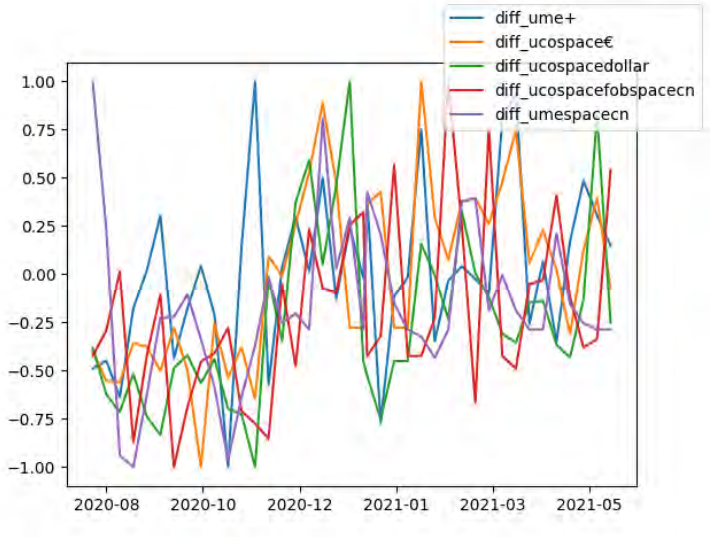


Figure 7: Weekly prices differences of UCO and UCOME variants.

done in Europe, as regulations there demand the use of biofuels for motorized road traffic.
UCO is a frequently traded and priced product on the biofuel market. From business expert knowledge, we know there is a dependency between prices for UCO and UCOME. Even more so, there is dependency between the different standardized versions of UCO or UCOME. In our data, we have five standardized products that receive regular price updates, as listed in Table (2). These prices have different freight terms. For FOB China freight terms, the UCO or UCOME is directly purchased in China whereas for CIF freight terms, the seller pays the freight cost and the buyer does not have to import the product himself. Since usually UCO comes from South East Asia and is shipped to Europe, the price differences between UCO with FOB China en CIF freight terms is approximately equal to the shipping costs of UCO from China to Europe.

Table 2: UCO and UCOME

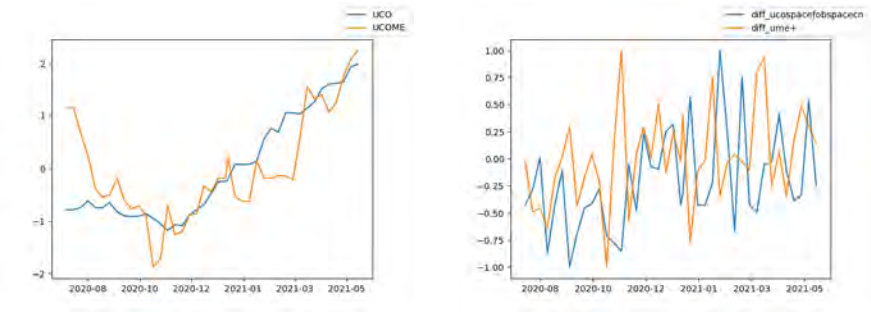
Name	Currency	Freight terms	Variable
UCO	EUR	CIF	ucospace€
UCO	USD	CIF	ucospacedollar
UCO	USD	FOB China	ucospacefobspacecn
UCOME	USD		ume+
UCOME	USD	FOB China	umespacecn

It is expected that the prices of these products move similar over time. Figure (6) shows the Spearman's correlation coefficient of the weekly differences of these prices. Figure (7) shows the scaled weekly price differences per product. From these figures we see the prices are indeed correlated to each other, especially the different variants within one product. Besides these there seems to be other factors and randomness that contributes to the price differences as well.

Table 3: Tickets

Name	Country	Category	Variable
THG C	Germany	Conventional	thgconventionspace21
THG O	Germany	Other	thgotherspace21
HBE C	the Netherlands	Conventional	hbeconventionalspace21
HBE O	the Netherlands	Other	hbeotherspace21

For the plant owner that produces UCOME out of UCO, costs will naturally rise when the price of UCO rises. Therefore, a increase in the UCO price might lead to a rise of the price of UCOME. Figure (8) shows the prices of UCO and UCOME. For comparison, the weekly prices in figure (8a) are normalized. We note that these normalized prices seem to follow the same trends over time. The weekly differences of these prices as shown in figure (8b) are therefore correlated as well.



(a) Normalized weekly price averages. (b) Differences of weekly averages.

Figure 8: UCO and UCOME assessment prices.

4.1.2 Tickets

Since tickets are used to fulfill emission obligations, they can substitute the use of biodiesel from the perspective of a fuel supplier. Therefore we expect the prices of these to relate in some way to the prices of biofuels. For a gasstation A that has to confirm with regulations, there are multiple options to do so. One option is to make sure that the diesel it brings into traffic, is a mix of traditional diesel and biodiesel. Another option to fulfill its obligation would be to buy emission rights. In that case, another gasstation B that blends more biofuel than it needs to confirm with regulations, can sell this surplus. This is done in the form of certificates or *tickets* that are registered by national authorities. In the Netherlands for example, this is the Nederlandse Emissie Autoriteit, the Dutch Emissions Authority. For a gasstation, buying tickets or blending biofuel are competitive options. When for example gasstations experience increasing costs for biodiesel, they might buy tickets instead, driving up the demand and therefore the price of tickets. It is therefore reasonable to expect that there exists a positive correlation between the prices of these tickets and the physical

biofuels. From Figure (9) we see that indeed the price of a ticket on the German market follows a somewhat similar trend as the prices of UCO and UCOME.

Figure 9: Normalized THG, UCO and UCOME price assessment.

4.1.3 Ticket categories

In the German market TreibHausGasquotes, THG's, are a way of trading obligations for blending biofuels. There are multiple variants of these tickets, based on the raw material that was used for producing the biofuel. In our data there are the categories Conventional and Other. Both variants can be used to fulfill an obligation. In order to stimulate the use of waste-based products as raw materials, biofuels produced using these materials generate a different ticket, the THG Other ticket. Both THG's can be used to fulfill an obligation. However, a minimum amount of the obligation should be fulfilled using the THG O ticket. Since this ticket has the advantage of counting towards this obligation, it has extra value compared to THG C tickets. Furthermore, in general waste based products are more expensive to produce. This factors make that usually the THG O ticket is slightly more expensive than the THG C ticket. Since there is only this price difference, the ticket prices will strongly correlate. In Figure (10), we see the THG prices over the period of August 2020 to May 2021.

4.1.4 Oil prices

In the biofuel industry, traders and brokers use the oil price and the movements of it as a reference. This means that when the oil price drops, biofuel prices will likely drop as well. Figure (11) shows the movement of the oil price and the UCO price assessment. For the oil price the price of a future on oil at the Nasdaq exchange is used. In order to compare the prices, both prices are divided by their mean over the total period.

4.2 Market participants and liquidity

Traders in the biofuel market often hold a portfolio of different products, which can both be physical biofuels as well as tickets. By the end of the year they have to make sure to fulfill their legal obligations in terms of blending biofuel

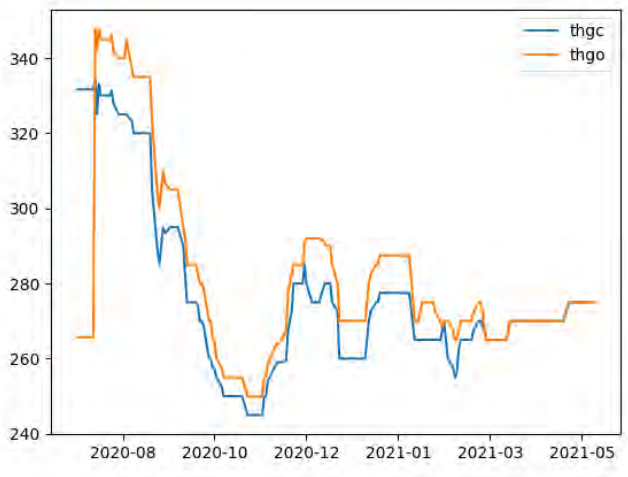


Figure 10: THG prices.

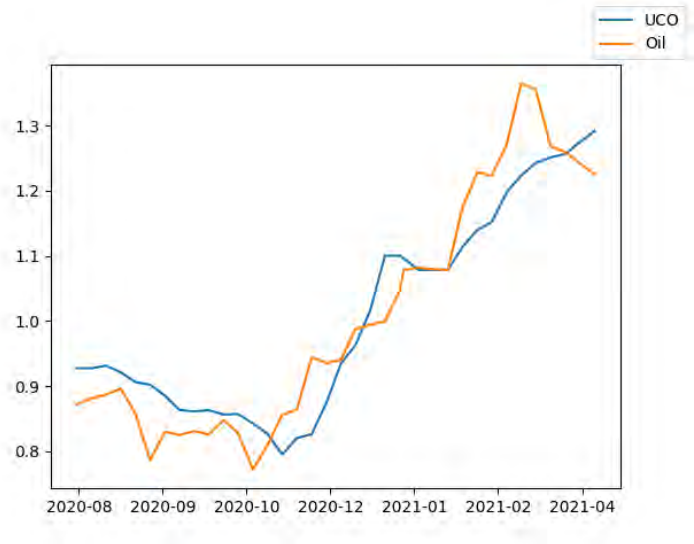


Figure 11: Weekly price of Nasdaq oil futures along with weekly average of price assessment of UCO.

or possessing enough tickets. Traders at these firms look for the right moment to sell, buy or exchange these products. Since these are mostly human actors, certain factors as expectations on the market play a roll. Brokers at Olyx observe this as well. This shows for example in a case where the price of THG tickets starts dropping. Many holders of these tickets were previously expecting the tickets would continue to rise for some more time. The unexpected drop in price changes the behavior of the traders, causing a lots of extra offers from sellers. As one may expect, this causes prices to drop even further because of a rise in supply. Buyers however do also take these opportunities, since it provides a nice moment to buy their obligations for a lower price, where they were only seeing increasing prices before. This behavior can be seen in the clustering of transactions, where there are some weeks with more than usual transactions. This is shown in Figure (12) that displays the number of ticket deals intermediated by Olyx per week. These described behaviour might not

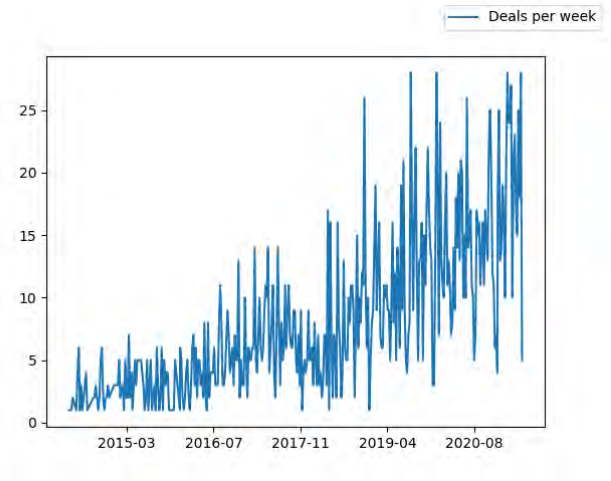


Figure 12: Number of ticket deals in German market per week.

always be rational in the sense that these decisions are supported by models that optimize a certain loss. Rather, these are the result of human behaviour. According to brokers however, this is how part of the market is responding. We could quantify this behaviour using metrics on the market activity, as for example the number of deals closed per week or the number of price updates per week. These might provide explanation in the movement of prices of biofuels.

4.3 Clustering

We would like to know whether the above described product dependencies, based on knowledge from business experts, is also apparent from the data. Figure (13) shows the Spearman pairwise correlation for the prices of the main dashboard products. This correlation coefficient assesses how well a relationship between two variables can be described by a monotonic, not necessarily linear, function. Clusters are displayed in this figure using lines, where lower level dependencies are between products. Higher order lines are dependencies between correlation groups of products or between a product and such a group. It is clear that the different biofuels and raw materials do indeed form a cluster. The tickets of different categories from the German market (THG Other, THG Conventional) are clearly correlated as well. This group also correlates with the prices of the tickets in the Dutch market (HBE).

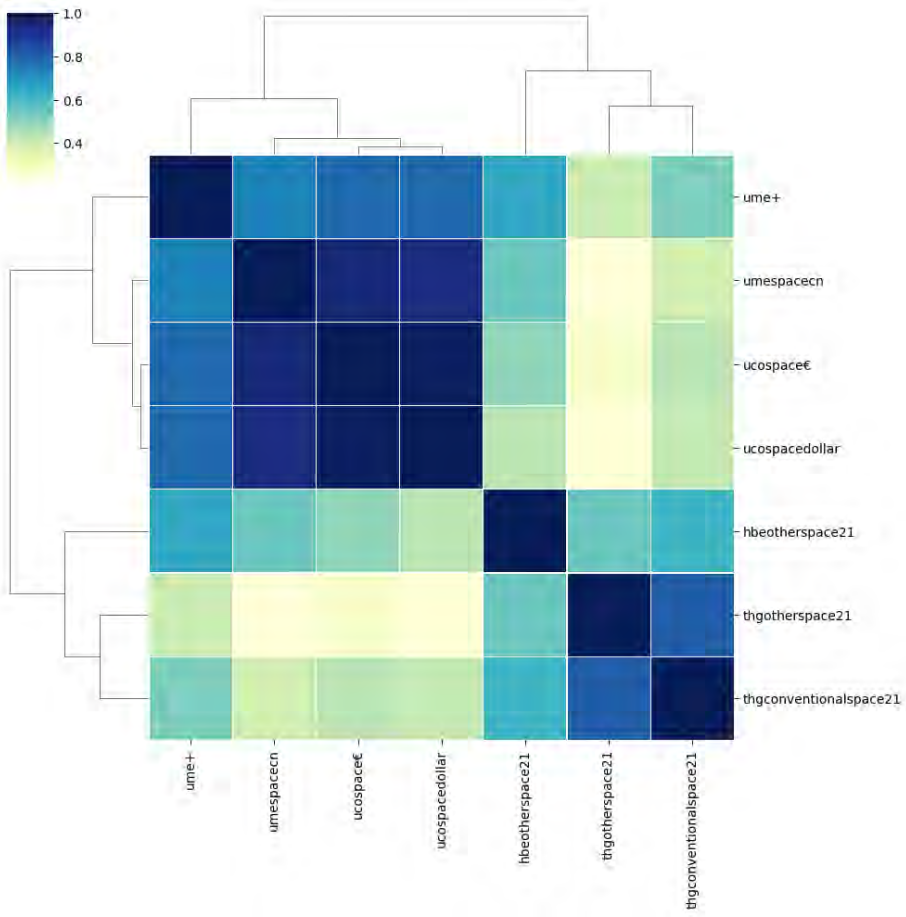


Figure 13: Clustermap with correlation between dashboard prices.

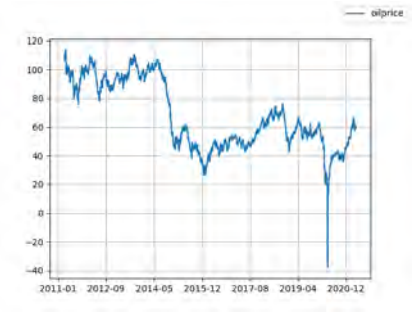
5 Experimental Evaluation and Results

In this chapter we model the relations between products as described in the previous chapters. We create a model and predict biofuel prices using an LSTM neural network. We make one step ahead predictions on the test set using these models and compare these to actual values. In Chapter 3 we already saw the model setup could learn trivial cases where the output prices completely depend on the input. In this chapter we build models for different groups of products. The LSTM models are in part defined by their hyperparameters. We discuss the improving of these models using different sets of hyperparameters and identify what hyperparameters have most significant impact.

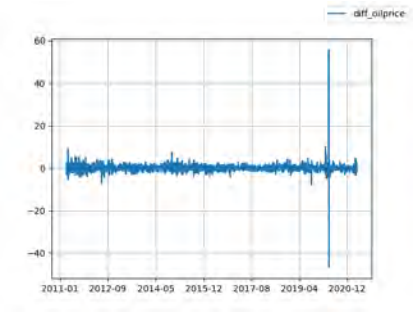
5.1 Single time series models

Oil price

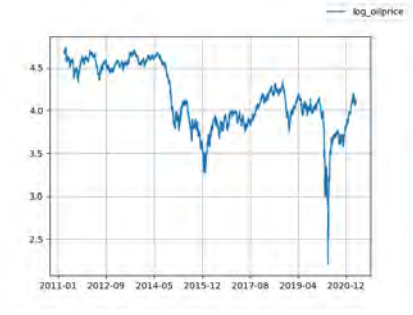
We investigate the Nasdaq oil price dataset with monthly prices from January 2011 to March 2021. The data is shown in Figure (14). From the data in Figure (14a), it is already clear the time series is not stationary. The first order differences of the oil price are shown in Figure (14b). We apply the logarithm transformation to the series of the prices, these are displayed in Figure (14c). The first order differences of the logarithm of the series, which can be interpreted as the return as discussed above, are shown in Figure (14d). The latter two time series where we took first order differences might be stationary, since from a first visual inspection we do not spot a clear trend in the data.



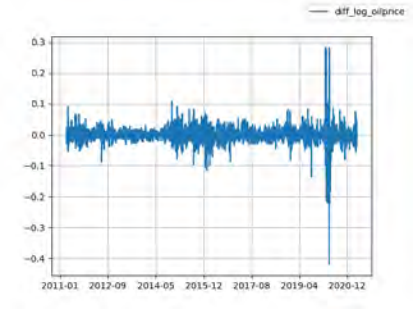
(a) Oil price.



(b) Daily differences oil price.



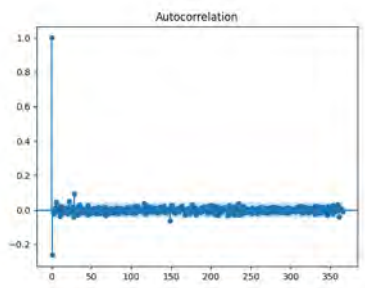
(c) Log oil price.



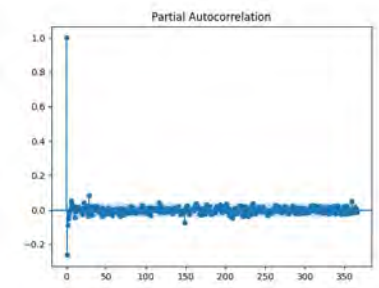
(d) Daily differences log oil price.

Figure 14: Oil price data.

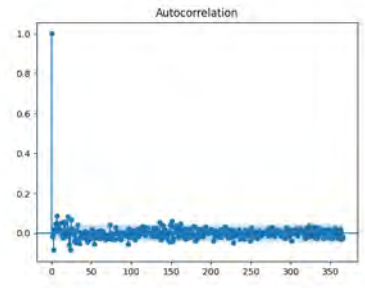
We study the autocorrelation and the partial autocorrelation functions of the time series. These are shown in Figure (15) and (16) for a lag of 365 days respectively 28 days.



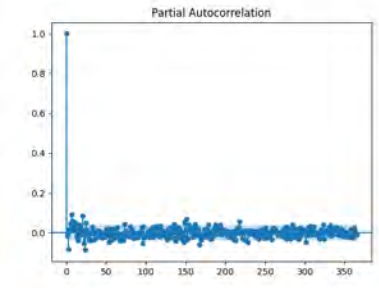
(a) ACF of first order difference of oil price.



(b) PACF of first order difference of oil price.



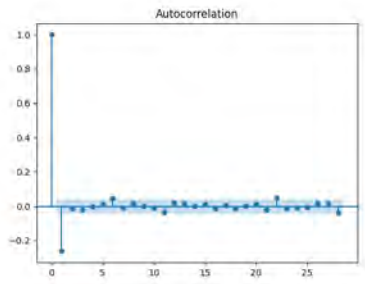
(c) ACF of first order difference of logarithm of oil price.



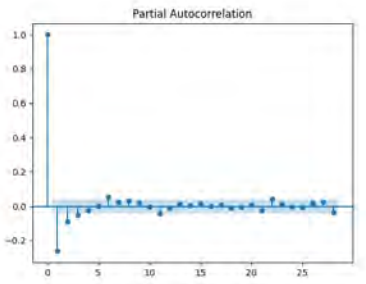
(d) PACF of first order difference of logarithm of oil price.

Figure 15: ACF and PACF of Oil price data, 365 lags.

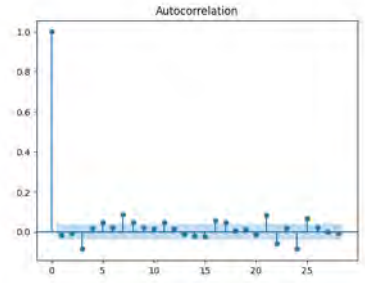
From the ACF and the PACF functions for the interval of 365 lags, we can conclude there is no clear periodic signal over a longer time period. This is the case for both the original time series as the differenced time series, as well as for the series where we applied the log transformation. When we look at the ACF and the PACF of the 28 days lag, we see in Figure (16a) and Figure (16b) there is a correlation in the differenced time series between X_t and X_{t-1}. Taking logarithms of the original time series and applying differences after that, seems to remove this correlation. This can be seen in the ACF and PACF in Figure (16c) and (16d).
The above suggests we may consider the differenced logarithms of the series as a stationary series. This is confirmed by the Dickey-Fuller test. For each of the three time series, we check for stationarity using the Dickey-Fuller test. The resulting p-values of these test are listed in Table (4). For both the differenced series and the differenced logarithms of the series, we can reject with $\alpha = 0.05$ the null hypothesis and assume the alternative hypothese that the series is stationary.
Now we know these series are stationary, we can estimate the parameters of the ARIMA model. We start with an ARIMA(1,1,1) model for the original series and the logarithm of the time series. This gives the following estimations as



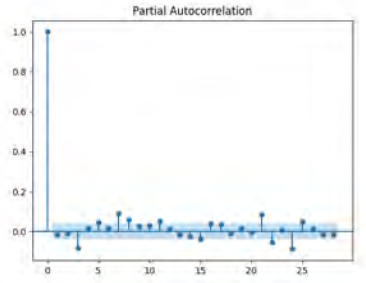
(a) ACF of first order difference of oil price.



(b) PACF of first order difference of oil price.



(c) ACF of first order difference of logarithm of oil price.



(d) PACF of first order difference of logarithm of oil price.

Figure 16: ACF and PACF of Oil price data, 28 lags.

Table 4: Test statistics and corresponding p-values of Dickey-Fuller test.

Time Series	Test Statistic	p-value
oilprice	-1.078	0.724
diff_oilprice	-15.249	0.000
diff_log_oilprice	-8.821	0.000

shown in Table (5) and Table (6). We show the estimates for the coefficients, the standard error, the z value and its corresponding p-value, the left and right boundaries of a central 0.95 confidence interval.

Table 5: Estimates of ARIMA(1,1,1) model on original series.

	Coef	S.E.	z	p-value	C.I. 0.025	C.I. 0.975
ar.L1	0.0735	0.018	4.159	0.000	0.039	0.108
ma.L1	-0.3634	0.017	-20.906	0.000	-0.398	-0.329
σ^2	3.5599	0.010 362.874	0.000	3.541	3.579	

Table 6: Estimates of ARIMA(1,1,1) model on logarithm of series.

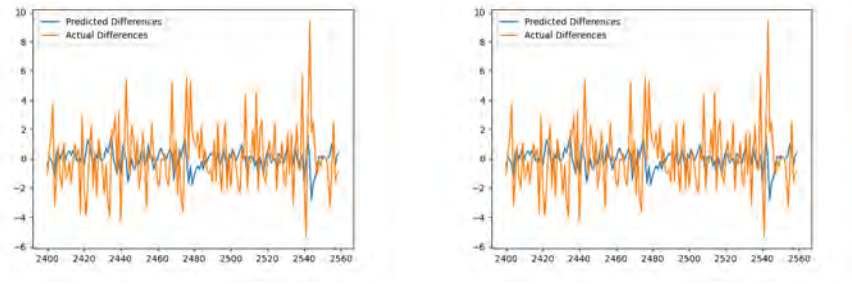
	Coef	S.E.	z	p-value	C.I. 0.025	C.I. 0.975
ar.L1	3.344e-06	0.003	0.001	0.999	-0.006	0.006
ma.L1	3.328e-06	0.003	0.001	0.999	-0.006	0.006
σ^2	0.0009	4.25e-06	203.122	0.000	0.001	0.001

For the model on the logarithm of the data we note that both the estimates for the autoregressive as the moving average part are very close to zero and do not have a significant p-value at $\alpha = 0.05$. Therefore, predictions for the differences of the log series made by this model will be equal to zero. This means if we transform back to the original time series, predictions for X_t will be equal to X_{t-1}. So this model is equal to the baseline model and thus does not improve on it.
For the model on the original data the estimations are shown in Table (5). These coefficients for the autoregressive and moving average part are not equal to zero and significant with $\alpha = 0.05$.
As a third step of our approach, we check the model found above is actually a good fit for the data. We do this only for the first model, where we found significant coefficients for the autoregressive and moving average part. The model does not seem to fit the data well. A subset of the predicted differences by the ARIMA(1,1,1) model are shown in Figure (17a), together with the the actual differences. The MSE of predicted differences compared to the actual differences is 3.56. The MSE of the baseline predictions is lower than this with a value of 0.301. We conclude that for both the model on the differenced series and the model on the logarithm of the differenced series, the historic prices in the ARIMA(1,1,1) model cannot provide extra information to make predictions on future values, other than the predictions based on the last known price.

5.2 Price prediction using product dependencies

LSTM

We start with a simple case where we use our knowledge of the current and historic price of a product X and the historic price of product Y, to predict the price of product Y at the current time. The price of tickets are a good choice for a first model, as these products are standardized and priced on a daily basis. We use the weekly differences of weekly averages of prices for the time series. We denote the weekly differences of prices of THG C tickets at time t with X_t,



(a) Predictions for differences.

(b) Predictions of differences of log.

Figure 17: ARIMA(1,1,1) model predictions.

the weekly differences of prices of THG O at t with Y_t. Suppose we observe $(X_1, \ldots, X_t)$, $(Y_1, \ldots, Y_{t-1})$. In the LSTM model we give as input

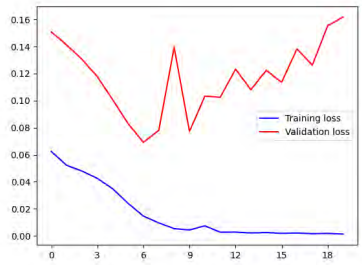
$$\begin{bmatrix} X_{t-l} & Y_{t-l-1} \\ X_{t-l+1} & Y_{t-l} \\ \cdots & \\ X_t & Y_{t-1} \end{bmatrix}. \tag{11}$$

As an output, we get a prediction for Y_t. The hyperparameters used for this experiment are listed in Table (7). The table also shows the MSE on the validation set, as well as the MSE of the baseline model on the same validation set as a reference. From both the comparison of these MSE's, as well as the fact that the MASE has value lower than 1, we can conclude that this model can predict the one step ahead future price.

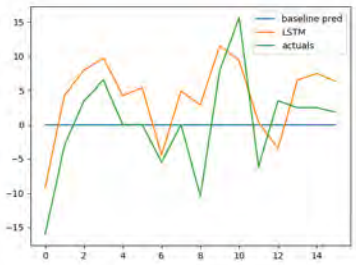
Table 7: Hyperparameters and results for the ticket experiment.

	Value
Sequence length	6
# epochs	20
Learning rate	0.010
Batch size	1
Train/val split fraction	0.500
Size training set	20
p	3
n	41
# nodes in LSTM layer	16
MSE on val. set	36.503
MSE of baseline on val. set	51.874
MASE	0.749
Features	diff_hbeother21, diff_thgother21, diff_thgconventional21

Figure (18b) shows the one step ahead predictions for the validation set. For comparison the actual values are shown as well as the baseline prediction.



(a) Loss per iterator over the network.

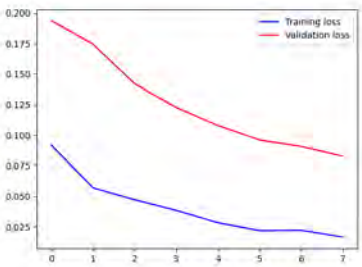


(b) Predictions for differences in daily averages by LSTM model and baseline model and actuals values.

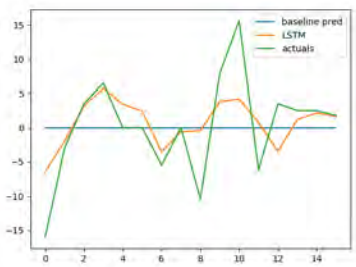
Figure 18: Ticket experiment 2.

Overfitting

Figure (18a) shows the loss on the predictions from the model during the training of the model. From this figure we see the model is overfitting on the training set. After about eight epochs, the loss on the validation set is increasing over the number of iterations of the network, while only the loss on the training set is decreasing. This suggests we might want to stop training earlier. In the next experiment, experiment 3, we run the same model, only with a lower number of iterations over the network. The results of this experiment are listed in Table (8). The loss throughtout the training process is shown in Figure (19a). The predictions of the model on the test set are shown in Figure (19b). Indeed the MSE is lower compared to previous experiment. The MSE is also lower than the MSE of the baseline model. Similarly, the MASE has a value lower than 1. This means the predictions of our model are more accurate than the predictions from the simple baseline model and may indeed be valuable.



(a) Loss per iterator over the network.



(b) Predictions for differences in daily averages by LSTM model and baseline model and actuals values.

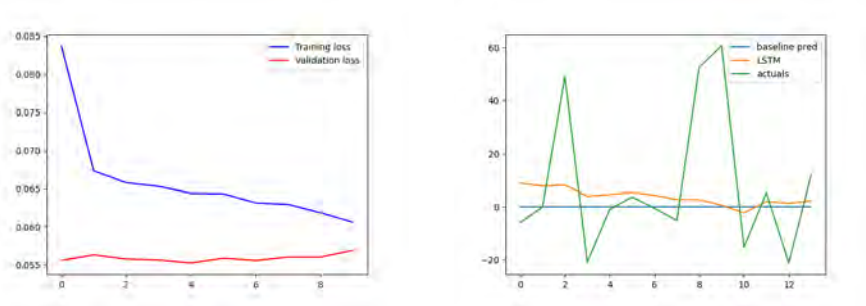
Figure 19: Ticket experiment 3 with early stop.

Table 8: Hyperparameters and results for the ticket experiment 3.

	Value
Sequence length	6
# epochs	8
Learning rate	0.010
Batch size	1
Train/val split fraction	0.500
Size training set	20
p	3
n	41
# nodes in LSTM layer	16
MSE on val. set	28.773
MSE of baseline on val. set	51.874
MASE	0.524
Features	diff_hbeother21, diff_thgother21, diff_thgconventional21

5.3 UCO & UCOME

In this experiment we model the price of UCOME using the price of UCO. We use weekly averages prices from the price assessment data for the different UCO and UCOME variants. Loss over the training of the network is shown in Figure (20a). The predictions of the model are shown in Figure (20b). Although the MASE of the model is somewhat below one, the model predictions seem not that accurate and very close to the predictions from the baseline model. Big changes in price that happened, where not at all predicted for by the model.



(a) Loss per iterator over the network.

(b) Predictions for differences in daily averages by LSTM model and baseline model and actuals values.

Figure 20: Ticket experiment with early stop.

Table 9: Hyperparameters and results for UCO & UCOME experiment 4.

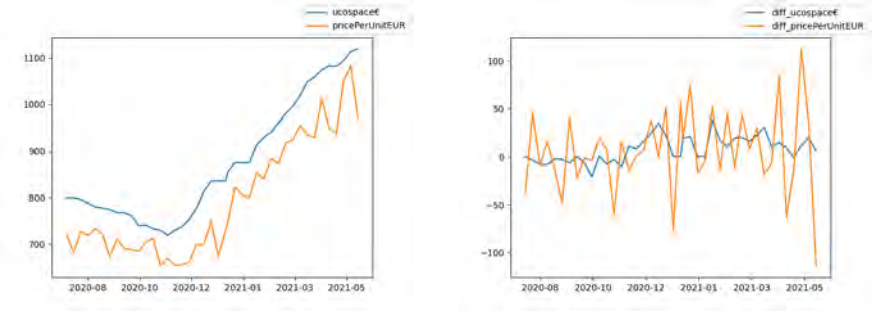
	Value
Sequence length	4
# epochs	10
Learning rate	0.010
Batch size	1
Train/val split fraction	0.600
Size training set	24
p	4
n	41
# nodes in LSTM layer	16
MSE on val. set	684.267
MSE of baseline on val. set	730.793
MASE	0.651
Features	diff_ume+, diff_uco€, diff_ucodollar

5.4 Deal

As discussed above, price assessments are made by brokers and indicate the approximate right price of a standardized product in the market at a certain time. These prices are however not directly tradeable. In this section we study whether price assessments do indeed contain information about market events.

5.4.1 UCO

We model the price of UCO deals using the price assessments. We use a similar model setup as in the previous section. An LSTM model is given the weekly averages of historic price assessments of relevant products, that is UCO and UCOME. Using this, the model predicts the average value of a metric ton of UCO as traded in a particular week.
Figure (21a) shows the price assessments of UCO together with the average price of actual traded UCO. It is clear from the figure that the prices do correlate. This is confirmed by the Spearman correlation coefficient between these series, that has a value of 0.946. The weekly differences of these prices are shown in Figure (21b). From a first visual sight, the similarity between these two time series is less clear. The Spearman correlation coefficient between these two differenced time series is 0.335, much lower than the correlation coefficient between the series themself. Our model, that predicts prices one week in advance, might therefore have trouble making accurate predictions.
The experimental setup and the results of the predictions for this experiment are listed in Table (10). The predictions are shown in Figure (22). It is clear that the price assessments do contain information about the deal prices, as expected. However, the relationship between these variables is not always direct, therefore one week ahead predictions are not always accurate. These predictions have a lower MSE than the baseline model. This means that the price assessments do contain extra information on top of the deal prices of last week.



(a) Average price of actuals trades per week and weekly average price assessments of UCO euro.

(b) Differences.

Figure 21: UCO prices: assessments and actuals.

Table 10: Experimental setup, hyperparameters and results.

	Value
Sequence length	2
# epochs	10
Learning rate	0.010
Batch size	1
Train/val split fraction	0.600
Size training set	24
p	6
n	40
# nodes in LSTM layer	16
MSE on val. set	1895.285
MSE of baseline on val. set	2166.563
Features	diff_ume+, diff_uco€, diff_ucodollar

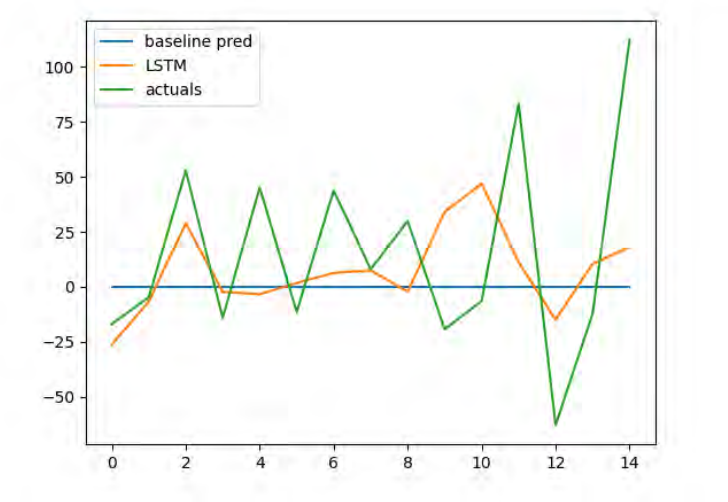


Figure 22: Predictions for deal prices.

5.4.2 Traded tickets

In this experiment we use the price assessments for the tickets to predict the prices of the German THG tickets as actual traded. The inputs of the model are the differences of the weekly averages of the market prices for tickets, listed as features in Table (11). The output of the model is the average weekly unit price of tickets as traded. Figure (23) shows the predictions and the actual values for the weekly differences. Table (11) shows the experimental setup, the hyperparameters used and the MSE of the predictions made. The MSE on the validation set is slightly lower than the MSE of the baseline predictions. This shows we can make predictions for the traded ticket prices that are more accurate than baseline predictions.

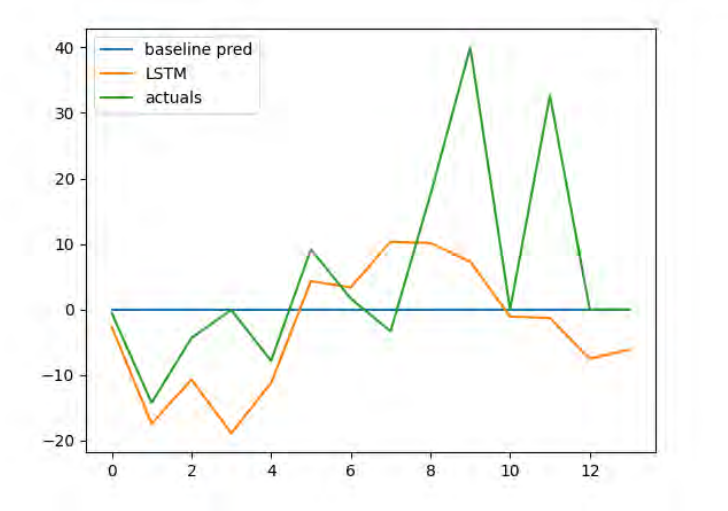


Figure 23: Predictions for deal prices.

Table 11: Experimental setup, hyperparameters and results.

	Value
Sequence length	4
# epochs	10
Learning rate	0.010
Batch size	1
Train/val split fraction	0.600
Size training set	24
p	4
n	41
# nodes in LSTM layer	16
MSE on val. set	214.561
MSE of baseline on val. set	239.562
Features	diff_hbeother21, diff_thgother21, diff_thgconventional21

5.5 Hyperparameter tuning

When building the LSTM model, several hyperparameters are used. These were already introduced in an earlier section where we discussed the layout of an LSTM network. Among these hyperparameters are the

- learning rate,
- nodes per layer and hidden layer size,
- number of layers,
- sequence length of the input data,
- number of epochs,
- batch size.

In their paper on LSTM models, Greff et al. [5] note that of these, the learning rate is the most important. The optimal value is dependent on the dataset. The hidden layers size is important as well and effects the performance of the network. In general, larger networks have better performance, but with diminishing returns [5].

5.5.1 Case 1

To study the choice of hyperparameters for our models, we take a model that was created in the previous subsection. Specifically we choose the model that predicts the ticket prices in the German market.
We vary the learning rate for the LSTM model with one hidden layer, since as noted above this hyperparameter is identified as most important. Figure (24) shows 1.) the time it takes to run to train the model and make predictions and 2.) the MSE of these predictions from the models with the different learning rates.
In the second setup we vary the number of epochs as well, next to the learning rate. The MSE and the execution time of the model with different learning rates and number of epochs are shown in respectively Figure (25a) and Figure (25b). It is clear that models with more iterations through the network, have a higher execution time. This differs from 5 seconds for only a few epochs to about 25 seconds for 20 epochs. The learning rate has little impact on the running time of the model. The performance of the model in terms of MSE however differs greatly for different hyperparamters settings. First, we observe that models with a higher learning rate (10^{-1}) have lower MSE than models with lower learning rate (10^{-6}). However, there is an interaction in this relation with the number of epochs used to train the model. Increasing the number of epochs gives a lower MSE, but this effect get stronger as the learning rate is smaller. For the highest learning rate (0.1) however, this relation between the number of epochs and the MSE is almost non existent. The interpretation for this is that models with a low learning rate learn slower. These models can however learn to make accurate prediction, but move slower to the prediction that gives lower MSE. Therefore, these models need a higher number of epochs to get to the estimation that will give predictions with lower MSE. The advantage that the slower models with low learning rate and high number of epochs could have, is

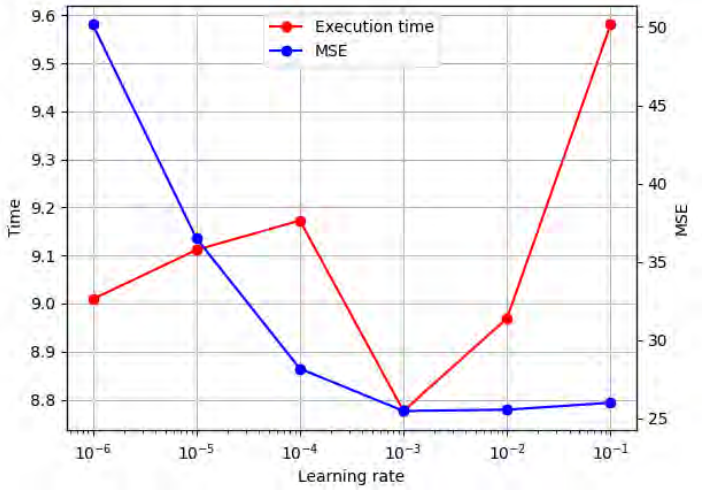
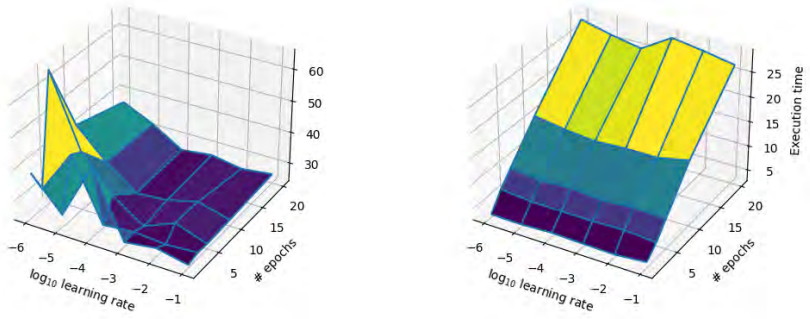


Figure 24: Execution time and MSE for varying learning rate.



(a) MSE (b) Execution time

Figure 25: MSE and execution time of models with varying learning rate and number of iterations through the network.

that they might find local minimum. Especially in non convex spaces, this can be powerful, as faster models with high learning rate might miss this solutions. This benefit comes then at the cost of higher execution time.

5.6 Deal regression

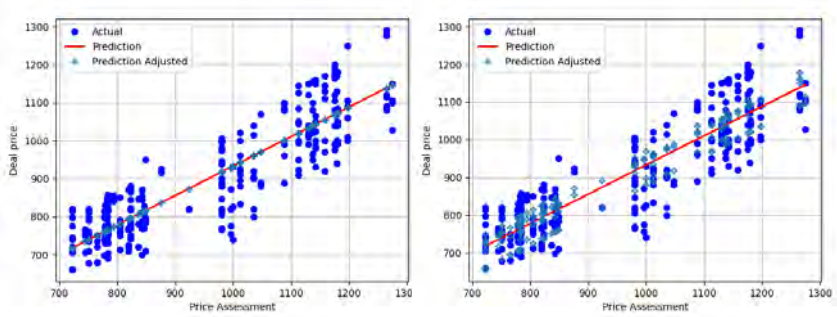
In this section we model the deal price of UCO. We use a regression model where the traded unit price of UCO is the dependent variable. The independent variables include the market price of UCO, as well as certain properties of the UCO product of the particular deal. We look at all the deals for UCO from 2014 until May 2021. The prices of these deals are expressed in euro per metric ton. Part of the deals are closed in USD, these prices are converted using the EUR/USD exchange rate on the day of the deal. In our model we do not consider the time that a deal is closed. Instead, we look at the weekly average of the price assessment of UCO as submitted by the brokers, in the week that the deal was closed.

5.6.1 Model 1

The first model takes the UCO price assessment as only independent variable. The estimate of the regression coefficient for the variable 'price assessment' and the coefficient of determination R^2 of the model are listed in Table (12). Figure (26a) shows the predictions as made by the model. We test whether the errors of the model are approximately normally distributed using Shapiro-Wilk test for normality. The test statistic is shown in Table (12) as 'errors normality' and with significance level $\alpha = 0.05$ we do not have to reject the null hypothesis that the errors follow a normal distribution. The explained variance of the model is relatively high with $R^2 = 0.785$.

Table 12: Regression model 1.

	Value
Errors normality	0.080
Price assessment	0.775
R^2	0.785

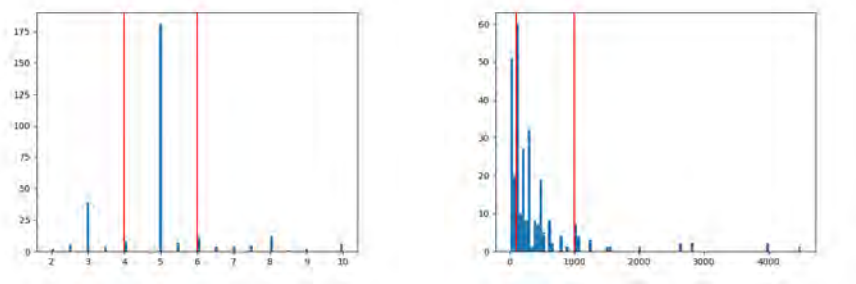


(a) Regression model 1. (b) Regression model 2

Figure 26

5.6.2 Model 2

To improve upon the model, we add extra independent variables. From expert knowledge we know that the FFA, a chemical property of the UCO, plays an important part in the unit price. The frequency of the different values for this variable in the sample are shown in Figure (27a). It is clear from the figure that the mode of the FFA values is 5. Furthermore there are some deals where the UCO has FFA lower than that, as well as higher FFA. Since this cannot be seen as a continuous variable, we choose to divide the FFA into three categories. UCO with FFA lower than four, FFA between four and six, and FFA higher than six. As a third variable in our model, we use the total volume of UCO of a deal. Figure (27b) shows the frequency of different volumes per deal. For the volume we make categories as well. We distinguish between low volume, medium volume and high volume deals. The intervals for these categories are respectively $[0, 100]$, $(100, 1000]$, $(1000, 10.000)$.
Adding the variables for the categories of FFA and the categories of the deal volume improves the model. Table (13) shows the results of this model. The regression coefficients for the price assessment and for each of the different FFA and volume categories are shown. Most notable the R^2 improves from our first model. This can also be seen by looking at the predictions directly. Figure (26b) shows the predictions from model two labeled as 'Prediction Adjusted'.



(a) Frequency of different FFA values, boundaries of the intervals of the categories are marked with a vertical red line.

(b) Frequency of different volumes, boundaries of the intervals of the categories are marked with a vertical red line.

Figure 27

Table 13: Regression model 2

	Value
Errors normality	0.195
Price assessment	0.789
FFA cat. 1	38.631
FFA cat. 2	21.494
FFA cat. 3	15.076
Volume cat. 1	-47.921
Volume cat. 2	22.855
Volume cat. 3	-2.820
R^2	0.828

5.7 Discussion

Conclusions from experimental evaluation

In this chapter we applied different model techniques to model and predict future differences of prices using the historic prices. We saw that these models can not make predictions for the future price difference of a product using only its historic price. In many cases however there are multiple products of which the prices correlate over time. Our models can make accurate predictions when we add to our model the current and historic prices of these related products. The prices of a single deal can be modelled using a relatively simple linear regression model. The market price has the most explanatory power in this model. Other factors that explain the price are the chemical properties of the fuel or raw material (the FFA) and the total volume of a deal.

Limitations

From the above experiments, there is an interesting observation that touches on a fundamental question on how we view our data. In Figure (22), we see for $t = 10$ that a rise in price is predicted. This rise however is only seen in the actuals at $t = 11$. In our experimental setup, this predictions leads to increase in MSE, since at both $t = 10, t = 11$, the prediction for the daily differences lay relatively far from the actual daily difference. However, one could also argue that our model at $t = 10$ foresees a rise in price that the markets only realize one week later at $t = 11$. From this perspective, it is not the model that was wrong. Rather, it is market participants that realize too late that the price should go up. This means that the model that is created might actually be valuable, but since this is not reflected in the MSE or MASE, this is not always recognizable. This also gives rise to questions on interaction between model and reality. In a future scenario, models as the ones described here may be used by different market participants. In that case, predictions made by the model may become reality because market participants act on them. This gives the models a self-fulfilling prophecy element. Especially in a market with relatively few transactions and limit number of participants, this effect may be quite strong. The models developed in this report are able to use input data in the form of time series. We mainly focused on data that is directly related to the biofuel market. Certain market trends however seem to be caused by other external factors, the massive price drop in the fuel and biofuel market in March 2020 as a clear example. Data sets that contain information on factors like this, could contribute to models. Further research may therefore focus on the question how to quantify events like these, for example using machine learning techniques that analyze news articles or sentiment analysis.

6 Conclusions and Recommendations

It is a challenging task to model and make accurate predictions on the prices of different biofuel products. By a lack of definite prices, price assessments and markets events are used to get as close as possible to a 'market price'. Our first research subquestion focuses on the prediction of prices using time series data:

- Can we predict biofuel market prices from historic price data in the form of time series only, using machine learning prediction models developed for time series?

We observe that most of these prices do not contain clear trends or seasonalities. ARIMA models have limited power in modelling these prices. Instead there are different direct and indirect relationships between the prices of different products. LSTM Neural Networks provide a good way to model part of these relationships. These models can be used to make one-step ahead predictions for time series. The resulting predictions are accurate and are able to model the direct relationships that exists between different products. These models however perform not as well when it comes to making predictions for the longer term. Longer term trends seem to depend on factors like future use of fuels and future regulations, which are due to their nature unavailable at the present. In order to create a good LSTM model, hyperparameters need to be set before training the model. Varying these may give models with more predictive power, that in addition also require less running time. The importance of different hyperparameters varies. The ones that have most significant influence on the performance of the model are the learning rate and the number of iterations through the network in the training process.
Our final subquestion regards the prediction of the price of a transaction:

- Can we predict the price of a product as it is actually traded between two parties as a function of 1.) the market price and 2a.) chemical properties of the product and 2b.) the product volume of the trade?

These prices of a single deal can be modelled using a relatively simple linear regression model. The actual market price has the most explanatory power in this model. The chemical properties of the fuel or raw material and the total volume of a deal have a significant influence on the price of a deal as well.
The modelling of price dependencies gives brokers insight in how prices relate to each other. These dependencies are often known, but not always directly manifest in the price. This leads to a discrepancy in prices that usually will be corrected later. With the use of modelling and predicting techniques, such differences can be detected earlier. This can lead to more accurate price information for the brokers and other market participants. This in turn will help brokers to make more accurate price assessments and therefore to provide market participants with more accurate price information. Similarly, deals can be closed at prices that fit the current market situation, as well as deals of similar products that have been closed in the past. Due to the delayed response of certain prices, models might predict price changes earlier than they actually occur. In our current setup, models are penalized for this behaviour, since predictions are compared to actual realizations. One could however argue that the actual price is inaccurate. Therefore it would be interesting to experiment with differ-

ent ways to account for this behaviour. As such, the score of models may more accurately represent their predictive power.
The models developed in this research are able to use input data in the form of time series directly related to the biofuel market. Certain market trends however seem to be caused by other external factors. Further research may therefore focus on the selection of other relevant datasets that contribute to price changes of products in the biofuel market.

References

[1] Bishop, Machine Learning and Pattern Recognition.

[2] Box, G.E.P., Jenkins, G.M., Time Series Analysis: Forecasting and Control, Fifth edition, 2016.

[3] Brownlee, J., *https://machinelearningmastery.com/time-series-forecasting-long-short-term-memory-network-python/*

[4] Dickey, D.A., Fuller, W.A., Distribution of the Estimators for Autoregressive Time Series With a Unit Root, Journal of the American Statistical Association, vol. 74, 1979.

[5] Greff, K., LSTM A search space Odyssey.

[6] Gunst, M.C.M. de, Statisical Models, January 2013.

[7] Hochreiter, S., Schmidhuber, J., Long Short Term Memory, Neural Computation 9, p. 1735-1780, 1997.

[8] Hyndman, R.J., Another Look At Forecast-Accuracy Metrics for Intermittent Demand, 2006.

[9] Namini. S.S., A Comparison of ARIMA and LSTM in Forecasting Time Series, *https://ieeexplore.ieee.org/abstract/document/8614252* , 2018.

[10] Olah, C., Understanding LSTM Networks *http://colah.github.io/posts/2015-08-Understanding-LSTMs/*

[11] Park, B., Bae, J.K., Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data, Expert Systems with Applications 42, 2015.

[12] Roy, A., Quantifying relationships between two time series data sets, 2016.

[13] Rumelhart, D.E., Hinton, E.H., Williams, R.J. Learning representations by back-propagating erros. Nature, vol. 323, October 1986.

[14] Sun, W., Huang, C., A carbon price prediction model based on secondary decomposition algorithm and optimized back propagation neural network, Jornal of Cleaner Production 243, 2020.

[15] Vaart, A.W. van der, Statistical Time Series.