ANOMALY DETECTION IN STOCK MARKET INDICES WITH NEURAL NETWORKS

Lucian Liviu Albu[1], Radu Lupu[1,2*]

[1]*Institute for Economic Forecasting - The Romanian Academy, Bucharest,Romania*
[2]*Bucharest University of Economic Studies, Bucharest, Romania*

Abstract

Neural networks have been long used for time series analysis in various applications. The late boost in computer power and data engineering brought about a myriad of algorithms that are wrapped under the larger title of Data Science. The apparent proliferation of these algorithms is due to their employment for several applications that range from simple classification problems, such as spam e-mail identification, to pattern detection in images and videos and several forecasting applications. Heralding the world of machine learning, these algorithms are trained on large amounts of data with the objective to extract repetitive structures that are likely to persist. It is therefore explainable the recent attention that these algorithms are given from the perspective of economic applications. This paper employs a recurrent neural network algorithm on daily data for several stock market indices in order to detect anomalous behaviour. The approach is rooted in the jump-detection literature that has the objective to identify outlying realizations of log-returns for diverse stock market data. We show that this approach establishes economically significant patterns that can be considered as anomalies when compared to their past dynamics.

Keywords: anomaly detection, neural networks, LSTM, stock market

JEL classification: C58; C53; G14

Introduction

Data science applications developed strongly in the recent years. Relying on a large literature developed during the 20th century and losing momentum for

[*] Corresponding author, **Lupu Radu** - radu.lupu@rei.ase.ro

some time until technology developed to allow for the most sophisticated algorithms, this field developed several tools that proved to be essential in several real-life applications, which is why it developed heavily in the field of computer science and engineering. The employment of probabilistic techniques which resemble the ones applied in econometrics, the newly developed tools serve for several tasks that could be categorized into classification problems, regression problems or supervised and unsupervised identification issues. The set of tools that captured momentum in the recent years is the one that encompasses the neural networks as algorithms that fare unexpectedly well in performing these tasks. It is therefore a very straightforward event the fact that these techniques touched the field of finance, which benefits from a very long history of data with random behaviour that could presumably feature changing patterns with time varying properties that persist in a nonlinear manner. This objective of this paper is rooted in this vein of research by challenging the common identification of abnormal returns via the literature of jump detection with a set of nonlinear models that are powered by neural networks specially designed for sequencing data such as time series. Looking at a set of data concerning stock market indices, we design a neural network model that captures the nonlinear dynamics existing in the data to identify non-normal or anomalous behaviour. We consider that this approach has the potential to enrich the current jump-detection paradigm by allowing for the possibility to detect more than just outlying data but also period of irregular movement of the respective time series. The usefulness of this approach relates to the general field of detecting systemic risk events, crashes that could root contagion phenomena in order to properly set stress test scenarios on one hand and to further investigate the possibility to develop early warning systems on the other hand.

1. Literature review

First of all, the financial data have some feature that make them special: a large amount and the lack of regularity. In this respect, Chen et al. (2019) suggested a technique to clean the data, appealing to a recurrently adaptive separation algorithm, more exactly to a maximal overlap discrete wavelet transform that allows for three actions concerning time-variant jumps, time-consistent patterns and marginal perturbations. After considering simulated and also US stock market data, the authors construed that the accuracy is improved for predictive models.

In financial research, diffusion models were used with the aim to emphasize the occurrence of jumps, but they have been shown not to be robust enough. Consequently, other methods were proposed and investigated. A mixture of well-known compound Poisson processes and Brownian motion model looks like has gained ground in finance and insurance, through the development of jump diffusion processes.

The machine learning algorithms were used by Au Yeung et al. (2020) to identify jumps in the financial time series; the authors combined this method with a type of Recurrent Neural Networks, namely Long short-term memory and derived a mixt method in order to detect anomalies that are not defined in advance in the model. This technique was applied on stock market data from 11 countries and the results were compared with those obtained from other methods like k-nearest neighbours algorithm, Hampel method, or Lee Mykland test, concluding that the new proposed method is better regarding the accuracy. The idea of stock specific characteristic for path dependence is promoted by M¨akinen et al. (2018) after applying a new method that is based on the Convolutional Long Short-Term Memory with Attention for five stocks from US market. They consider that their results are better than those obtained after using the multi-layer perceptron network or the Long Short-Term memory model.

The contribution of Hawkes in the field of jumps (selfexciting or mutuallyexciting Hawkes processes) is found in the 70s, but recently their application in the financial field as "contagious jumps" was acknowledged, according to Hawkes (2020).

The technological progress contributed to the availability of Big Data, expansion of globalisation and also changed the industrial phase. Through the applications available due to the development of artificial intelligence (especially machine learning techniques), anomalies can be detected in different economic systems. A history of research conducted in recent years is presented by MATEOS GARC´IA (2019). The role of anomaly detection is essential for every economic/industrial activity, banking system or network disturbance, but they are treated differently depending on the size of the data. For the case of big size data, Ramchandran and Sangaiah (2018) proposed a mixt setting for the unsupervised anomaly detection algorithm, alleged DBN-K that allows for real-time identification.

2. Data and methodology

Our analysis relies on daily data for ten stock market indices collected for the period between January 2000 and September 2020. We developed an investigation that consisted in a sample of approximately 5300 observations for the following indices: AEX (AEX Index from the Amsterdam Stock Exchange), ASX (S&P/ASX 200 Index from the Australian Securities Exchange), ATX (ATX Index from the Austrian Stock Exchange), BEL (BEL 20 Stock Index from the Euronext Brussels), CAC (the benchmark for the French stock market index), DAX (the German stock market index), HSI (Hang Seng Index from Hong Kong), IBEX (the Spanish stock market index), NIKKEI (the Japanese stock market index) and SP500 (the US stock market index). Statistical properties of log-returns and information about the sample size for each of this indices in provided in Table 1. The variation is sample sizes is due to bank holidays that are different in each national stock exchange.

Table no. 1: Statistical properties of log-returns for stock market indices

	count	mean	std	min	25%	50%	75%	max
AEX	5352	-3.2E-05	0.013983	-0.11376	-0.00603	0.000437	0.006476	0.100283

ASX	5299	0.000132	0.010263	-0.10203	-0.00449	0.000424	0.005305	0.067665
ATX	5253	0.000114	0.014348	-0.14675	-0.0063	0.000392	0.007381	0.12021
BEL	5349	-3E-06	0.012692	-0.15328	-0.00551	0.000255	0.006237	0.09334
CAC	5347	-3.4E-05	0.014406	-0.13098	-0.00656	0.000253	0.007114	0.105946
DAX	5319	0.000122	0.014864	-0.13055	-0.0066	0.000657	0.007386	0.107975
HSI	5206	0.000065	0.014495	-0.13582	-0.00657	0.000222	0.007176	0.134068
IBEX	5345	-9.9E-05	0.014689	-0.15151	-0.00711	0.000415	0.00716	0.134836
NIKKEI	5219	0.000042	0.014784	-0.12111	-0.00686	0.000074	0.007759	0.132346
SP500	5232	0.000167	0.012573	-0.12765	-0.0048	0.00059	0.005795	0.109572

Source:Bloomberg, author’s calculation

2.1. Autoencoder

Our analysis consisted in the use of recurrent neural networks in order to detect possible anomalous dynamics in these indices. The main engine in our model is the autoencoder, which is a particular kind of neural network employed in order to perform unsupervised learning of data codings. The main feature of this construction is to achieve a reduction of dimensionality by means of converting the data with the objective to eliminate the noise.

This process of data representation is accompanied by another one that pursues its reconstruction from the encoded version. The simplest version of an autoencoder is a network that contains an input layer, a hidden layer and an output layer of neurons. This construction attempts to ensure that data is copied from the input layer to the output layer as accurate as possible with the constraint that the hidden layer extracts only the most relevant aspects of the data. In this way we may say that the hidden layer delivers the actual coding (a description of the code) of how to eliminate the irrelevant part of the data. A representation of a simple autoencoder is depicted in Figure no. 1.

The general objective of replicating the data from the input side to the output side requires the implementation of an approximation. Starting to be used as means of dimensionality reduction the concept is widely employed as part of artificial intelligence developments which place this encoders inside deep neural networks to achieve higher accuracy.

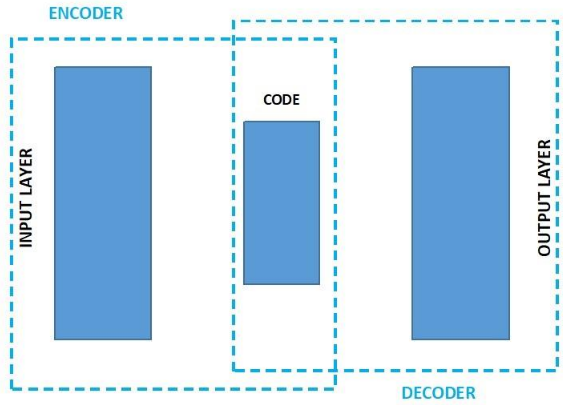


Figure no. 1: A simple Autoencoder

Source: author's conceptualization

2.2. Recurrent neural network - RNN

The recurrent neural networks are algorithms that are developed based on several layers of neurons (units) with the specific purpose to address machine learning problems that consist in sequential data. They are therefore designed to capture the complexity of temporal sequencing and they also own an internal memory that helps in addressing the long-term persistence. The basic RNN is organized into ordered layers of neurons such that each layer is connected to every node in the following layer. Each such neuron contains an activation that is time dependent and all connections have changing weights that modify as data flows through the network. RNNs have the capacity to update their current state as function of the past states and the new input data. There are many attempts to configure architectures for RNN, which are nowadays used and developed into more complex algorithms (Elman (1990), Jordan, Chen et al. and Sterˇ (2012)). An important breakthrough took place with the development of specific neural networks designed to handle ”long-term dependencies”. Hochreiter and Urgen Schmidhuber (1997) created the long short-term memory networks, also known as LSTM. Due to its powerful learning capacity, LSTM networks are widely used in several tasks such as natural language processing and time series. A thorough review of LSTM cells and network architectures is created by Yu et al. (2019).

2.3. LSTM Autoencoder

Prediction problems started with the case of predicting the following value from a sequence of data, which requires the development of a forecasting algorithm that can fall into the topic of ”many-to-one”. However, most important issues arise when we deal with the prediction of several moments in the future based on a series of data, which is known as sequence-to-sequence prediction problems (seq2seq).

The main concern here is development of an neural network that should have the ability to use a certain size for the input sample and produce a different size of the output sample, a problem which is know as ”many-to-many”. According to Cho (2006), the encoder-decoder LSTM architecture is designed to capture seq2seq prediction problems. As the usual autoencoder, this algorithm consists in two models: one for parsing the input sample and encoding it into a vector and another one for decoding the vector and generating the predicted values (output). Confronted with an application that concerns the video representation, Srivastava et al. (2015) mentions the employment of LSTM encoder-decoder as presented in Figure no.1.

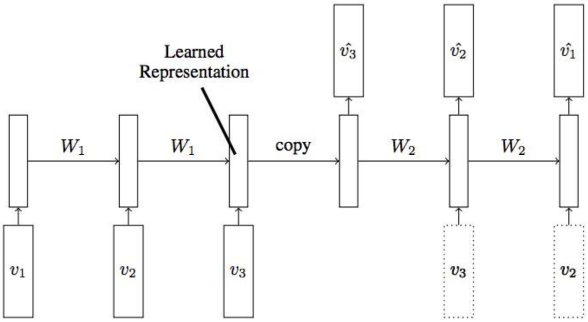
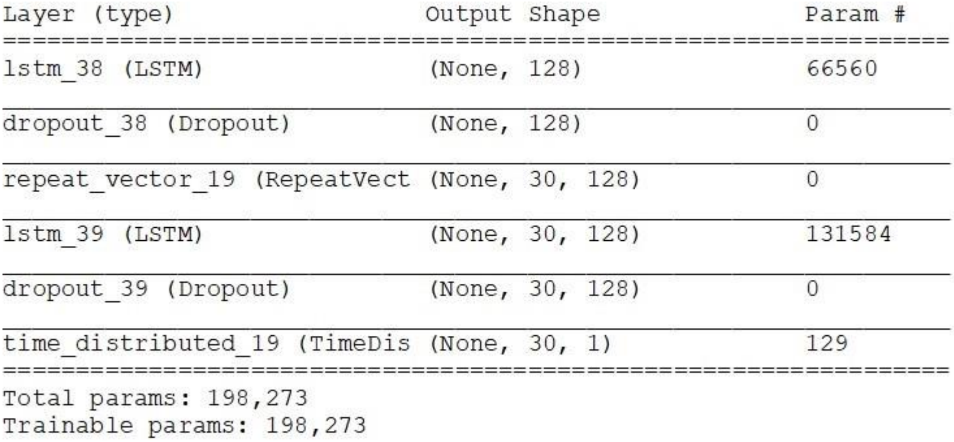


Figure no. 2: LSTM Autoencoder model as presented by Srivastava et al. (2015)

Source: Srivastava and colab. (2015)

We use this principle to create a LSTM autoecoding architecture that allows for the analysis of log-returns for our stock market data. This encompases an LSTM architecture with 128 neurons for the encoder, a dropout and a repeat vector and another LSTM network with 128 neurons for the decoder. The data was sequenced into samples of 30 observations each. There are 198,273 parameters that this autoecoder calibrates during the learning process, as presented in Figure no 3.



```
Layer (type)                 Output Shape              Param #
=================================================================
lstm_38 (LSTM)               (None, 128)               66560
_________________________________________________________________
dropout_38 (Dropout)         (None, 128)               0
_________________________________________________________________
repeat_vector_19 (RepeatVect (None, 30, 128)           0
_________________________________________________________________
lstm_39 (LSTM)               (None, 30, 128)           131584
_________________________________________________________________
dropout_39 (Dropout)         (None, 30, 128)           0
_________________________________________________________________
time_distributed_19 (TimeDis (None, 30, 1)             129
=================================================================
Total params: 198,273
Trainable params: 198,273
```

Figure no. 3: LSTM Autoencoder structure replicated for each series (extraction from Python algorithm)

Source: author's conceptualization

3. Results and discusions

Fitting the LSTM encoder provided results structured in the following charts. The model described in the previous section was trained up to 100 epochs (although usually the training finished in less than 20 epochs on average) in which we monitor the training loss and the validation loss. We used 10% of data for validation set. We used a patience parameter of 3 to verify for the extent to which we reached a plateau or we found the global minimum of validation loss. Running this algorithm for each of our data set rendered a set of in-sample errors for our calibrations, which we depict in Figure no. 4.

The main interesting observation is that the distributions depicted in this chart (Figure no.4) are rather similar, with almost the same scale on both the horizontal and vertical axes. They are all skewed to the right, which means that there are situations where our returns tended to be less captured by the autoencoder and may result in possible anomalous developments. We notice larger spikes for the case of ASX, ATX, NIKKEI and SP500. If we consider our analysis as an investigation into the extent to which the past dynamics tend to be repeated in the future, we can say that there are certain patterns in the future, usually not in large size, that keep a certain level of uncertainty, which tends to be similar for each of our series. The employment of neural networks is notoriously considered important for time series analysis and forecasting due to its non-linear ability to capture repetitive patterns. The training sample consisted in all data until January 1st 2010 and the test set contains all the data from January 1st until September 15th 2020. This means that the algorithm learns from a behaviour that covered the main stock market crises from the beginning of 2000 and the big financial crisis from 2007 and 2008.

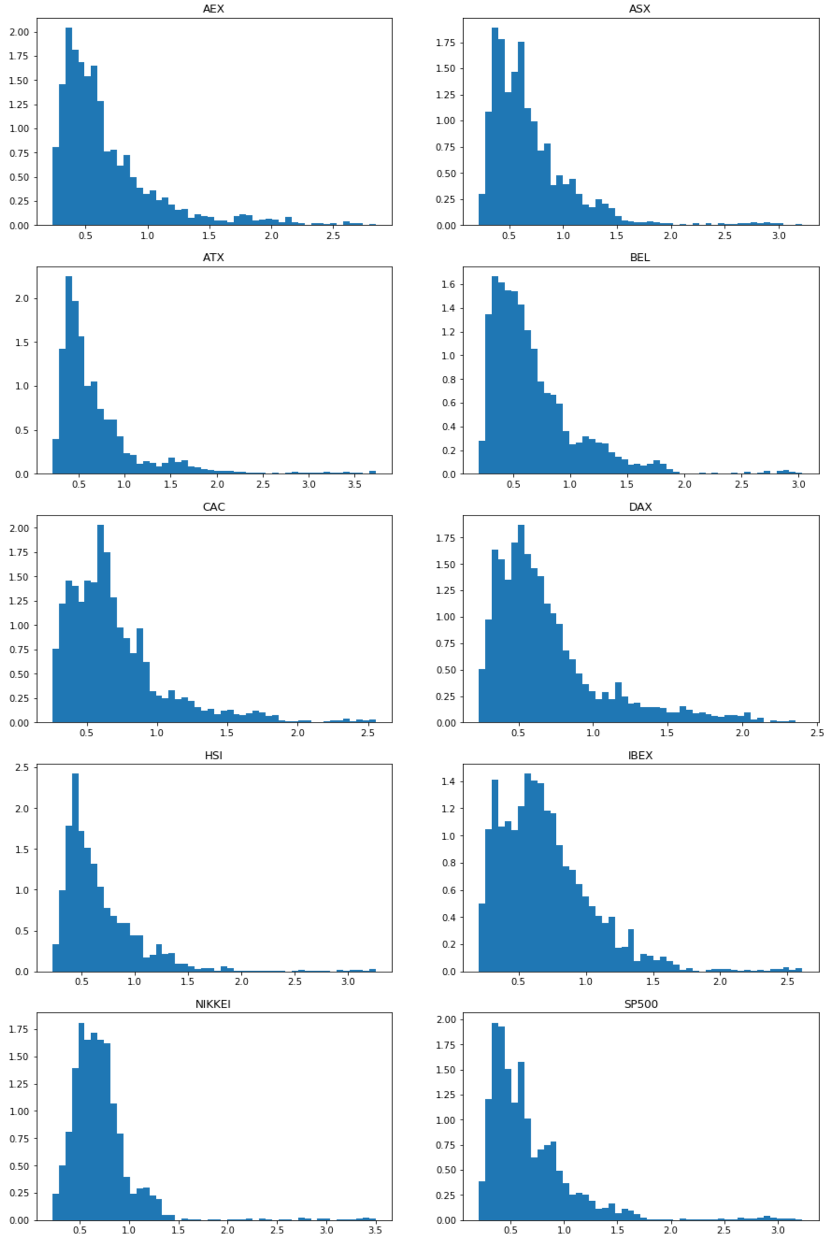


Figure no. 4: Histograms of training errors (in-sample) for each series of log-returns

Source: author's conceptualization

We can say that in this way the neural network has the chance to experiment several abnormal phenomena with complex nonlinear changes. Based on the set of parameters inferred from this train data we employed the specification of the neural network to generate forecast for the testing data set. The set of errors obtained for each observation is grouped into histograms as depicted in Figure no.5. We can notice again the fact that, with few exceptions, the set of errors for the test set have similar distributional shapes, with the same skewness as the training errors and similar scales for the two axes. We notice that the larger horizontal axes belong to ASX, ATX and SP500.

We also note that none of these distributions has multiple modes, which is a simple tool that provides evidence that the algorithm succeeds to capture the existing patterns with feasible accuracy. Given this ability to capture repetitive patterns in the data, we are using this algorithm to extract anomalous dynamics in the log-returns, which could be expressions of systemic risk manifestations in the data. Our attempt to spot these anomalies relies on this calibration to settle a threshold that will help us separate the regular errors from the anomalous ones. We decided to estimate this threshold based on the distribution of errors in the train sample (in-sample analysis) by choosing the 95% quantile. Figure no. 5 depicts the histograms of errors in the test set (out-of-sample) and the thresholds that were obtained by using this procedure. We notice that there will be a situation where the threshold is too high, which is the case for HSI log-returns. In this case we can say that the behaviour of the train set was more anomalous than the one in the test set, which means that the second part of the data series behaved in a more normal manner.

Based on this proposed methodology we were are now able to identify the actual moments in time when our algorithm spotted the non-regular dynamics. A representation of these moments is generated in Figure no. 6. With only few exceptions we can notice that the anomalies of the log-returns were detected mainly in the COVID-19 time frame, i.e. in March 2020, when the start of the pandemic crisis impacted the markets. The large number of anomalies detected in this period can be seen as a confirmation of the accuracy of the proposed algorithm. In this respect, we can argue that the market reaction in this time frame is the most extreme one, resembling the large crisis from 2007 and 2008. An important element of these results is the fact that the algorithm picks up anomalies only in March, which is clearly consistent with the general market sentiment. This means that in the period post March, stock markets returned to a rather normal type of dynamics, reflecting expectations for a ”new-normal” phase.

The largest accumulation of abnormal returns was captured in the case of ASX, BEL, NIKKEI and especially SP500. We notice that these correspond to the situations where the log-returns were the larges (both negative initially and positive afterwards) when compared to the rest of the stock indices under analysis. Shocks were also captured in other periods for the ASX in 2016, ATX also in 2016, IBEX in about the same time as the previous two and mostly for NIKKEI in the same time. We observe that these abnormal behaviour picked up by our algorithm is consistent with the realization of large returns during that time., but not necessarily in all the situations when we had these large negative or positive returns, which is another interesting effect. When comparing this anomaly detection process with a simple jump detection algorithm, we need to state the fact that we are not looking for simply situations where returns are large but also to the general abnormal dynamics of a particular time stamp. If the jump detection literature attempts to detect situations when log-returns reached higher levels as compared to the local volatility (by this keeping track of the current market conditions), our approach is able to identify a period with potential disruptive dynamics, which could be determined by the longer impact of influential events on the stock prices.

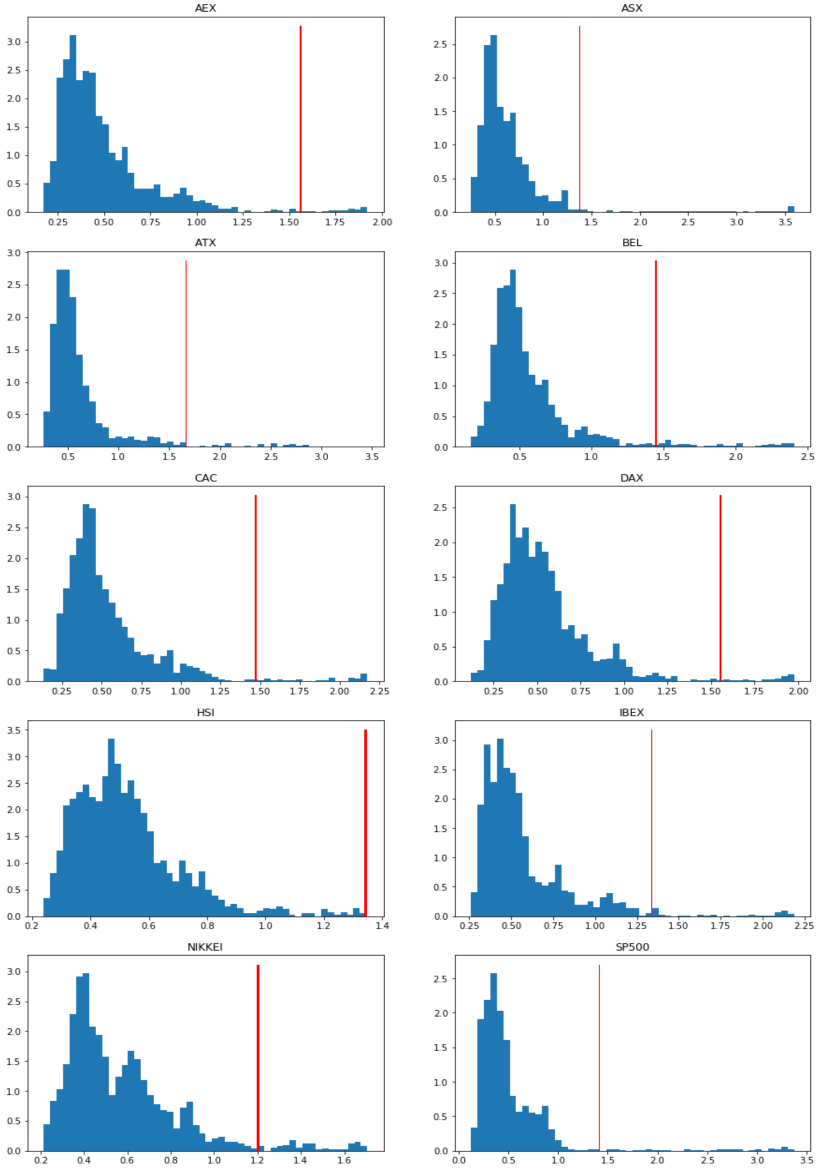


Figure no. 5: Histograms of testing errors (out-of-sample) for each series of log-returns

Source: author's conceptualization

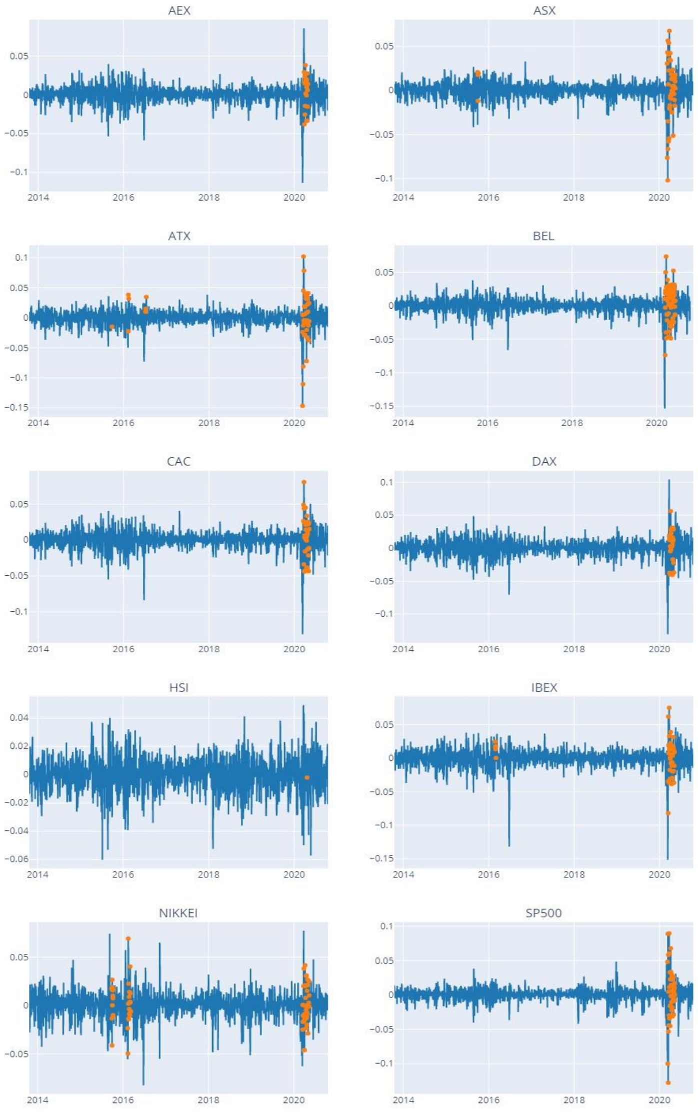


Figure no. 6: Log-returns and Anomalies detected with LSTM Autoencoder

Source: author's conceptualization

Conclusion

This paper uses daily data for a set of ten stock market indices for a twentyyear period to develop an analysis on the possibility to detect anomalies in log-returns. Each data set is analysed separately with an LSTM autoencoder and we separate the training set and the testing set according with respect to the same time moment (January 1st 2010). Our objective was to understand the utility of the non-linear data characterization provided by neural networks on time financial time series so that we may be able to obtain more than just extreme returns as previously developed by the literature of jump-detection. Our results show that this approach offers interesting perspective on the application of such machine learning techniques. It is not only that we succeeded to identify anomalous behaviour but we were also able to understand that such behaviour is related to clear influential events, such as the burst of the pandemic crisis in March 2020. Due to these results we consider that the approach presented in this paper deserves further attention to be used as tool for systemic risk detection and creates the opportunity for further study of early warning algorithms.

References

[1] Alan G. Hawkes. Hawkes jump-diffusions and finance: a brief history and review. *European Journal of Finance*, 2020. ISSN 14664364. doi: 10.1080/1351847X.2020.1755712.

[2] Anitha Ramchandran and Arun Kumar Sangaiah. Unsupervised Anomaly Detection for High Dimensional Data - an Exploratory Analysis. In *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*, pages 233–251. Elsevier, 1 2018. doi: 10.1016/B978-0-12-813314-9.00011-6. URL https://linkinghub.elsevier.com/retrieve/pii/B9780128133149000116

[3] Branko Ster.ˇ Selective Recurrent Neural Network. *Springer*, 2012. doi: 10.1007/s11063-012-9259-4.
URL https://www.researchgate.net/publication/257631827.

[4] Jay F.K. Au Yeung, Zi kai Wei, Kit Yan Chan, Henry Y.K. Lau, and Ka Fai Cedric Yiu. Jump detection in financial time series using machine learning algorithms. *Soft Computing*, 24(3):1789–1801, 2 2020. ISSN 14337479. doi: 10.1007/s00500-019-04006-2.

[5] Jeffrey L. Elman. Finding Structure in Time. *Cognitive Science*, 14(2):179–211, 3 1990. ISSN 03640213. doi: 10.1207/s15516709cog14021.
URL http://doi.wiley.com/10.1207/s15516709cog1402$_1$.

[6] Kyunghyun Cho. Learning Phrase Representations using RNN En-coder–Decoder for Statistical Machine Translation Kyunghyun. *Journal of Biological Chemistry*, 281(49):37275–37281, 2006. ISSN 00219258.

[7] Michael I. Jordan. Attractor dynamics and parallelism in a connectionist sequential machine - Artificial neural networks.
URL https://dl.acm.org/doi/abs/10.5555/104134.104148.

[8] Milla M¨akinen, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Forecasting of Jump Arrivals in Stock Prices: New Attention-

based Network Architecture using Limit Order Book Data. 10 2018. URL http://arxiv.org/abs/1810.10845.

[9] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised Learning of Video Representations using LSTMs. *32nd International Conference on Machine Learning, ICML 2015*, 1:843–852, 2 2015. URL http://arxiv.org/abs/1502.04681.

[10] Nuria MATEOS GARC´IA. Multi-agent system for anomaly detection in Industry 4.0 using Machine Learning techniques. *Adcaij: Advances In Dis-Tributed Computing And Artificial Intelligence Jour-Nal*, 8(4):33, 9 2019. ISSN 2255-2863. doi: 10.14201/adcaij2019843340.

[11] Sepp Hochreiter and J J Urgen Schmidhuber. Long short-term memory. Technical Report 8, 1997. URL http://www7.informatik.tu-muenchen.de/hochreit http://www.idsia.ch/ juergen.

[12] TB Chen, VW Soo Conference on Neural Networks (ICNN'96), and undefined 1996. A comparative study of recurrent neural network architectures on learning temporal sequences. *ieeexplore.ieee.org*. URL https://ieeexplore.ieee.org/abstract/document/549199/.

[13] Yi Ting Chen, Wan Ni Lai, and Edward W. Sun. Jump Detection and Noise Separation by a Singular Wavelet Method for Predictive Analytics of High-Frequency Data. *Computational Economics*, 54(2):809–844, 8 2019. ISSN 15729974. doi: 10.1007/s10614-019-09881-3.

[14] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: Lstm cells and network architectures, 7 2019. ISSN 1530888X. URL https://www.mitpressjournals.org/doi/abs/10.1162/neco$_{a0}$1199.