FACHHOCHSCHULE NORDWESTSCHWEIZ

MASTER THESIS PROPOSAL

Thesis Title

Author:
Kevin SANER

Supervisor:
Prof. Dr. Thomas HANNE

A thesis proposal submitted in fulfillment of the requirements for the degree of Master of Science in Business Information Systems

in the

School of Business

April 24, 2021

Declaration of Authorship

I, Kevin SANER, declare that this thesis titled, "Thesis Title" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"Your brain does not manufacture thoughts. Your thoughts shape neural networks."

Deepak Chopra

FACHHOCHSCHULE NORDWESTSCHWEIZ

Abstract

School of Business

Master of Science in Business Information Systems

Thesis Title

by Kevin SANER

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too...

Acknowledgements

The acknowledgments and the people to thank go here.

Contents

List of Figures

List of Tables

List of Abbreviations

DNN	**D**eep **N**eural **N**etwork
CNN	**C**onvolutional **N**eural **N**etwork
RNN	**R**ecurrent **N**eural **N**etwork
GRU	**G**ated **R**ecurrent **U**nit Network
LSTM	**L**ong **S**hort **T**erm **M**emory

For/Dedicated to/To my…

Chapter 1

Introduction

With the rise of the Internet of Things (IoT) and ever more sensors, gadgets and smart devices like smartwatches for fall detection or blood pressure monitoring, or fridges for temperature protective control in use, the amount of available data steadily increases (Alansari et al., 2018). Simultaneously, the possibilities to use the data to draw conclusions increases. This data is generally used to draw conclusions such as failure of a system or a medical issue, such as a heart attack. These events typically occur very rarely (Hauskrecht et al., 2007). However, when the number of instances of each class is approximately equal, most machine learning algorithms function best. Problems occur when the number of instances of one class greatly exceeds the number of instances of the other. This issue is very popular in practice, and it can be observed in a variety of fields such as fraud detection, medical diagnosis, oil spillage detection, facial recognition, and so on (Thabtah et al., 2020). The task of identifying the rare item, event or observation is often referred to as anomaly detection. Typically, the anomalous item translates to problems such as bank fraud or medical problems. Often, the anomaly does not adhere to the common statistical definition of an outlier. Therefore, many outlier detection methods (in particular unsupervised methods) fail on such data (Hodge and Austin, 2004)

A special disciplin in anomaly detection is to find the anomaly in a time series. The anomaly detection problem for time series is usually formulated as finding outlier data points relative to some standard or usual signal. Time series anomaly detection plays a critical role in automated monitoring systems. It is an increasingly important topic today, because of its wider application in the context of the Internet of Things (IoT), especially in industrial environments. The most popular techniques to find the anomalies are:

- Statistical Methods
- Support Vector Machines
- Clustering
- Density-based Techniques
- Neural Networks

This research paper purely focusses on Neural Networks for anomaly detection. Especially Recurrent Neural Networks and Convolutional Neural Networks are investigated and compared.

1.1 Definitions

Following, the most important terms in the context of anomaly detection using neural networks are elaborated and defined.

1.1.1 Univariate, Bivariate and Multivariate Data

Time series data comes in different forms. It is distinguished between univariate, bivariate and multivariate data. Univariate involves the analysis of a single variable while bivariate and multivariate analysis examines two or more variables.

Univariate Data

There is only one variable in this type of data. Because the information only deals with one variable that changes, univariate data analysis is the simplest type of analysis. It is not concerned with causes or relationships, and the primary goal of the analysis is to describe the data and identify patterns.

Bivariate Data

This type of data involves two different variables. This type of data analysis is concerned with causes and relationships, and the goal is to determine the relationship between the two variables.

Multivariate Data

Multivariate data is defined as data that contains three or more variables. It's similar to bivariate, but there are more dependent variables. The methods for analyzing this data are determined by the objectives to be met. Regression analysis, path analysis, factor analysis, and multivariate analysis of variance are some of the techniques.

An example of a multivariate time-series is the collected data from several sensors installed in a car. One main difference between time-series and other datasets is that the observations do not only depend on components d, but also on the time feature n. Thus, time-series analysis and the used statistical methods are mostly different from the methods used for random variables that assume independence and constant variance of the random variables. To data analysts, time-series are important in a variety of fields like economy, healthcare and medical research, trading, engineering and geophysics. These data are used for forecasting and anomaly detection.

1.1.2 Neural Networks

An Artificial Neural Network (ANN) with several layers between the input and output layers, is known as a Deep Neural Network (DNN). Neural networks come in a variety of shapes and sizes, but they all have the same basic components: neurons, weights and functions. These components work in the same way as human brains and can be trained just like any other machine learning algorithm.

Neuron

Artificial neurons represent the smallest building blocks of neural networks. A neuron usually receives separately weighted inputs which it sums. The sum is then passed through the activation function to calculate the output of the neuron. When training a neural network, the input weights are adjusted by the optimizer function to improve accuracy of the given task e.g. classification.

Layer

In neural networks three different kind of layers are distinguished. There are input, output and hidden layers. A layer can be described as a collection of neurons. All layers between the input and output layer are called hidden layers. In the input layer data is fed into the neural network. The output of the hidden layer is calculated by taking the weighted sums of input and passing it through the activation function. Typically, a more complex problem requires more hidden layers to accurately calculate the output. In the output layer the final result e.g. a classification is produced. Figure 1.1 how a simple Neural Network with just one hidden layer could look like.

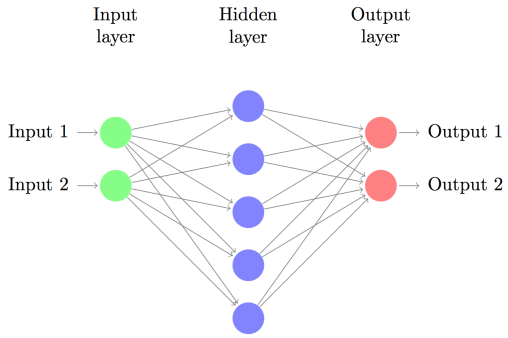

FIGURE 1.1: Input, Hidden and Output Layers (Denny Britz, 2015)

Optimizer Function

1.2 Background

Following, it is explained how different kinds of neural networks work and what they are used for.

1.2.1 Neural Networks for Anomaly Detection

Out of the three most popular neural network architectures, convolutional neural networks (CNN), recurrent neural networks (RNN) and deep neural networks (DNN), only RNN are typically used for anomaly detection in time series. RNNs have built-in memory and are therefore able to anticipate the next value in a time series based on current and past data. Classic or vanilla RNNs can theoretically keep track of arbitrary long-term dependencies in input sequences. There, however, exists a computational issue: when using back-propagation to train a vanilla RNN, the back-propagated gradients can "vanish" or "explode" due to the computations involved in the process, which use finite-precision numbers. Because LSTM units allow gradients to flow unchanged, RNNs using LSTM unit or Gated Recurrent units (GRU) partially solve the vanishing gradient problem and therefore drastically improve accuracy.

Specially to mention in this context are LSTM (Long-Short Term Memory) and GRU (Gated Recurrent Units). Both achieved outstanding performance when used

for tasks such as unsegmented, connected handwriting recognition, speech recognition and anomaly detection in network traffic or IDSs (intrusion detection systems) (Junyoung Chung, 2014)

LSTM

LSTM was first proposed in 1997 by Schmidhuber and Hochreiter (Hochreiter and Schmidhuber, 1997). The initial version to the LSTM unit consisted of a cells, input and output gates. In 1999, the LSTM architecture was improved by introducing a forget gate and therefore allowing the LSTM to reset its own state (Gers, Schmidhuber, and Cummins, 2000). LSTM is used in a supervised training approach, that means it tries to predict a predefined state taking the past and the current state. If the predicted state differs from the expected state, the weights of the different gates are adjusted using an optimizer algorithm such as gradient descent. Figure 1.2 shows how the gates and the cell are arranged. The cell represents the memory of the LSTM. In simple words, the LSTM works as follows to predict a new value:

1. Forget Gate: Obsolete information is removed from the cell state.
2. Input Gate: New information is added to the cell state
3. Output Gate: The new information and the cell state are added to make the new prediction.
4. The new cell state is propagated to the next LSTM unit

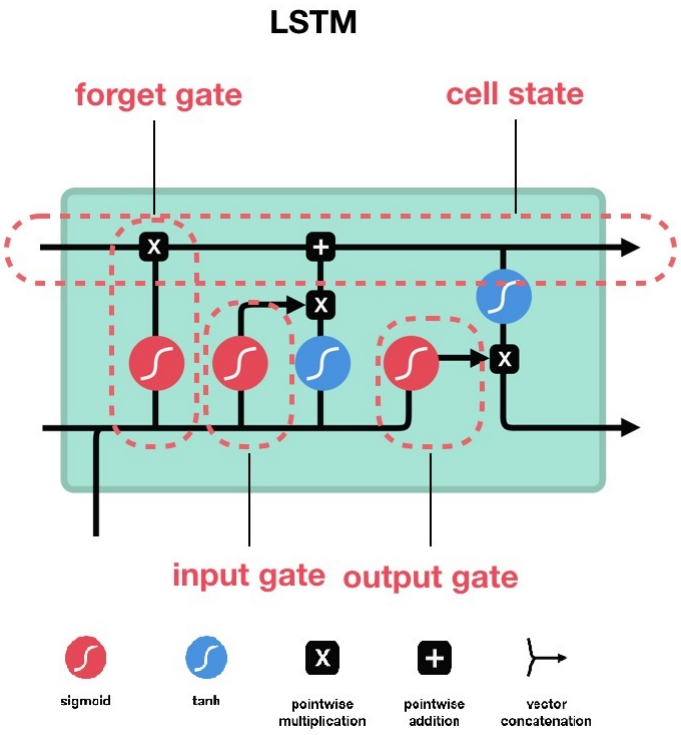

FIGURE 1.2: Gates and Cell of LSTM (Michael Phi, 2018)

CNN

In contrast to RNNs Convolutional Neural Networks are generally used for image classification. CNNs work as feature extractors and are able to recognize patterns. CNNs use layers that are not fully connected, to reduce complexity (compare to 1.2.1). In a CNN, a set number of neurons forms a filter. These filters or kernels are the actual feature extractors. A filter may represent a line or pattern (see Figure 1.3).(Aggarwal, 2013)

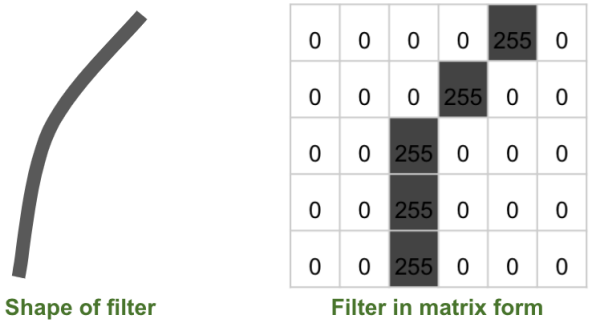

FIGURE 1.3: Example of a Filter used in CNN

To detect whether, a feature is occurrent in a picture, the filter is gradually moved over the picture in so called strides. In every step (stride) the dot-product between the filter and the part of the picture is calculated. The results of the operations are stored in activation maps. The greater the dot-product the more alike are the filter and the section of the image. Training the network hereby refers to determining the shapes of these filters. Other typical features of a CNN are the pooling layers. The pooling layers reduce the amount of computation necessary. The most commonly used pooling technique is max-pooling and works as shown in Figure 1.4.

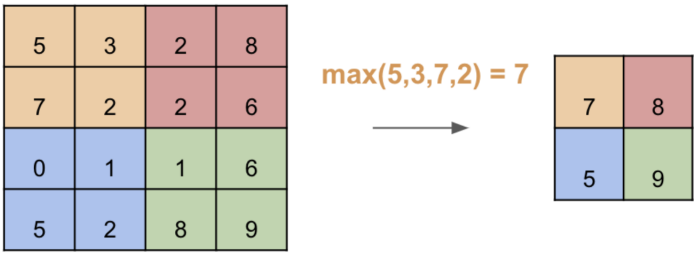

FIGURE 1.4: Example of max-pooling

The idea of max-pooling is to only keep the maximum value of an activation map. In the orange region 7 represents the maximum value, so it is kept while the other values are discarded (Rich Stureborg, 2019).

In 2019, Wen and Keys proposed to use CNN also for anomaly detection in time series since it shares many common aspects with image segmentation. A univariate time series is therefore viewed as a one-dimensional image.

1.2.2 Transfer Learning

The reuse of a previously trained model on a new problem is known as transfer learning. It is currently very popular in deep learning because it can train deep neural networks with a relatively small amount of data. This is particularly useful in the field of data science, as most real-world problems do not provide millions of labeled data points to train complex models. In transfer learning, the knowledge of an already trained machine learning model is applied to a different but related problem. For example, if a classifier was trained to predict if an image contains a backpack, the model's experience could be applied to recognize other objects such as sunglasses (Niklas Donges, 2020).

1.3 Problem Statement

Defining a ground truth is one of the most difficult aspects of time series anomaly detection. Determining when anomalous behavior begins and ends in time series is a difficult task, as even human experts are likely to disagree in their assessments. Furthermore, there is the question of what constitutes a useful detection when detecting anomalies in time series. In the past, RNN have successfully been used for anomaly detection (e.g. [Malhotra et al., 2015; Kim et al., 2016; Wang et al., 2017; Yin et al., 2017]. Therefore, designs for various use cases are well researched. RNN are well suited for the task, however, take a long time to train due to the complexity of how a single unit is designed (see 1.2.1). In comparison CNN are not as complex and therefore, generally take less time to train. However, CNNs are generally used for image recognition and were only very recently used for anomaly detection in time series. It is therefore mostly unknown what designs are applicable for successful anomaly detection in time series data. While RNN are able to deal with multivariate data by design, a classical CNN requires design changes to be able to deal with multivariate data. Wen and Keys (2019) proposed to use a special kind of U-net, an improved version of a Fully Convolutional Neural Network (Ronneberger, Fischer, and Brox, 2015). Further, a CNN is not capable to analyze streaming data so it relies on segmentation of the data. These data segments are called snapshots. In order to not miss any data points, the frequency of taking these snapshots should be at least as high as the length of snapshot so that every time point is evaluated by the model at least once. However, for better performance it might prove beneficial to use a higher frequency which means every point is evaluated various times by the model (Wen and Keyes, 2019). The proposed design change and the fact that every point is evaluated multiple times, increases complexity and evaluation time and therefore counteract the architectural advantage of CNN compared to a RNN. When designing a neural network many parameters have to be chosen, this applies to both mentioned types of Neural Networks. For example, when designing a CNN, the number of layers, the activation function(s) of a single neuron and the optimizer function have to be chosen. Additionally, when using CNN for time series data the length and frequency of the snapshots have to be determined. Similarly, when designing a RNN also the number of layers and the optimizer function have to be determined. Because the basic building blocks of both networks types are very different it is difficult to fairly compare the complexity of two architecture approaches. Another important parameter which applies to both network types is the number of epochs for which the networks are trained. Through the epochs the training time is determined. In order to compare the two types of neural networks, two networks of similar complexity have to be designed. With equal training time the performance of

both can be compared and evaluated. A RNN is therefore only set up as benchmark while the main goal of this research project is to clarify whether CNNs are really useful and propose an advantage over RNNs when applied on time series data for anomaly detection.

1.4 Thesis Statement

Convolutional Neural Networks are superior to Recurrent Neural Networks when looking for anomalies in time series data regarding training time and complexity.

1.4.1 Subquestions

- How does a CNN for univariate and multivariate data need to be designed for successful anomaly detection in time series data?
- What advantages and disadvantages arise when using a CNN compared to a RNN for anomaly detection in univariate and multivariate time series?
- What parameter settings are crucial for a fair performance comparison between RNN and CNN?
- Optional: How does transfer learning affect the performance of CNN compared to RNN in anomaly detection in time series?

1.4.2 Research Objectives

Following the research objectives of this paper are defined.

1. Determine what design changes a CNN requires to detect anomalies in time series data.
2. Determine how the CNN should be designed for the comparison with a RNN
3. State the advantages and disadvantages of the chosen CNN architecture.
4. Define parameters which allow a fair comparison of CNN and RNN

1.4.3 Limitations

Recently there have been approaches that combine CNN and RNN into a hybrid network for tasks such as handwriting recognition or video-based emotion recognition (Dutta et al., 2018) (Fan et al., 2016). However, this paper only compares pure CNN and RNN, and does investigate a hybrid approach.

1.4.4 Significance

Until now, time series data was almost only approached with RNNs. This paper should answer the question whether CNNs propose a valid alternative and even propose some advantages over RNNs. The paper will answer the fundamental question whether research should channel efforts to further investigate CNNs for anomaly detection in time series data or whether no benefits can be discovered and research is better to focus on other areas.

1.4.5 Chapter Overview

Chapter 2

Literature Review

2.1 Anomaly or Outlier?

Generally, there is no agreement on how to distinguish between anomalies and outliers. The following often used citation proves equality of the term outliers and anomalies.

"*Outliers are also referred to as abnormalities, discordants, deviants, or anomalies in the data mining and statistics literature.*" - Aggarwal (2013)

By others, outliers are regarded as corruption in data, while anomalies are abnormal points with a particular pattern. In the context of this paper, only the term anomaly is used to refer to such irregular behaviour. It is hereby important to provide a clear definition for the concept of anomaly. This is critical since different meanings of abnormalities necessitate different detection methods. As a result, it is important to identify the key characteristics of anomalies and to use the description to highlight the boundaries. Following, two of the most common definitions of anomalies:

"*Anomalies are patterns in data that do not conform to a well defined notion of normal behavior.*" - Chandola et al. (2009)

Ord (1996), defines anomalies as follows:

"*An observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data.*"

Anomalies have two major features, according to both of these definitions:

- The anomalies' distribution deviates significantly from the data's overall distribution.
- Standard data points make up the vast majority of the dataset. The anomalies make up a very small portion of the overall dataset.

The development of anomaly detection methods is dependent on these two factors. The second property, in particular, prevents the employment of common classification methods that depend on balanced datasets.

2.1.1 Types of Anomalies

Anomalies come in a variety of shapes and sizes. Anomalies can be divided into three categories:

1. **Point Anomalies** - A point anomaly occurs when a single point deviates dramatically from the rest of the data. A point anomaly is, for example, a large credit transaction that differs from other transactions.

2. **Collective Anomalies** - Individual points may or may not be anomalous, but a series of points may be. A bank customer, for example, withdraws $500 from her account per weekday. Although withdrawing $500 every now and then is common for the consumer, a series of withdrawals is unusual.

3. **Contextual Anomalies** - Some points can appear natural in one context but be identified as anomalous in another: In Germany, a daily temperature of 35 degrees Celsius is considered natural in the summer, but the same temperature in the winter is considered unusual.

Knowing ahead of time what kind of anomaly the data might contain helps the data analyst choose the best detection process. Some methods for detecting point anomalies fail to detect collective or contextual anomalies entirely (Braei and Wagner, 2020).

2.1.2 Time Series Patterns

There are a few key characteristics of time-series that are briefly described here.

Level

The mean of the series is used to determine the time-series standard. When a time-series has a pattern, the level is also said to be changing.

Trend

If the mean of a time series does not remain constant over time but increases or decreases, it is said to have a trend. A pattern can be either linear or non-linear in nature. Figure 3 shows a positive trend from 2005 to 2008, and then a downward trend after that.

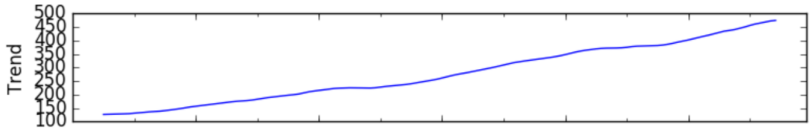

FIGURE 2.1: Trend ()

Seasonality

Seasonality refers to the occurrence of variations on a regular basis. Seasonal variables such as the time of year, day of the week, and other similarities influence the time-series, which is why it is called seasonal. As a result, it has a set period of time that is often limited to a year. A seasonal time-series is depicted in Figure 4 (Braei and Wagner, 2020).

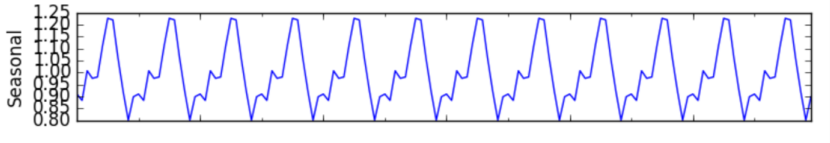

FIGURE 2.2: Seasonality ()

Noise

The variability in the observations that the model cannot account for.

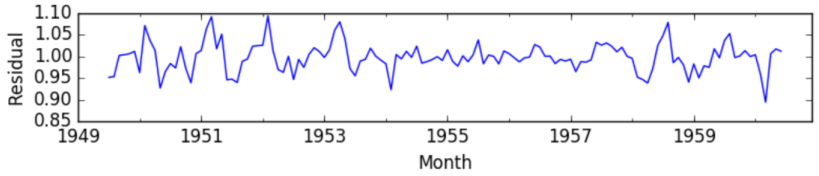

FIGURE 2.3: Noise ()

Observed

All the above componenents combined could provide the observed time series shown in Figure . The components may add up to form a model such as:

$$Y = level + trend + seasonality + noise \tag{2.1}$$

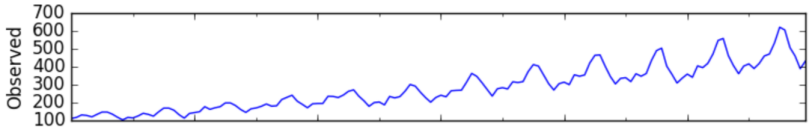

FIGURE 2.4: Observed ()

2.2 RNN for Anomaly Detection

– state of the art

2.2.1 CNNs

2.2.2 CNN for time series data

2.2.3 Parameter Settings for fair comparison

2.2.4 Transfer Learning

2.3 Research Gap

Bibliography

Aggarwal, Charu C. (2013). *Outlier analysis*. Vol. 9781461463962. DOI: 10.1007/978-1-4614-6396-2.

Alansari, Zainab et al. (2018). "The rise of Internet of Things (IoT) in big healthcare data: Review and open research issues". In: *Advances in Intelligent Systems and Computing*. Vol. 564. DOI: 10.1007/978-981-10-6875-1_66.

Braei, Mohammad and Sebastian Wagner (2020). *Anomaly detection in univariate time-series: A survey on the state-of-the-art*.

Chandola, Varun, Banerjee, Arindam and Vipin Kumar (2009). "Anomaly detection: A survey". In: *ACM computing surveys (CSUR)* 41.3, pp. 1–58.

Denny Britz (2015). *Implementing a Neural Network from Scratch in Python – An Introduction*. URL: http://www.wildml.com/2015/09/implementing-a-neural-network-from-scratch/ (visited on 04/18/2021).

Dutta, Kartik et al. (2018). "Improving CNN-RNN hybrid networks for handwriting recognition". In: *Proceedings of International Conference on Frontiers in Handwriting Recognition, ICFHR*. Vol. 2018-Augus. DOI: 10.1109/ICFHR-2018.2018.00023.

Fan, Yin et al. (2016). "Video-Based emotion recognition using CNN-RNN and C3D hybrid networks". In: *ICMI 2016 - Proceedings of the 18th ACM International Conference on Multimodal Interaction*. DOI: 10.1145/2993148.2997632.

Gers, Felix A., Jürgen Schmidhuber, and Fred Cummins (2000). "Learning to forget: Continual prediction with LSTM". In: *Neural Computation* 12.10. ISSN: 08997667. DOI: 10.1162/089976600300015015.

Hauskrecht, Milos et al. (2007). "Evidence-based anomaly detection in clinical domains." In: *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*. ISSN: 15594076.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long Short-Term Memory". In: *Neural Computation* 9.8, pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: https://direct.mit.edu/neco/article/9/8/1735-1780/6109.

Hodge, Victoria J. and Jim Austin (2004). *A survey of outlier detection methodologies*. DOI: 10.1023/B:AIRE.0000045502.10941.a9.

Junyoung Chung Caglar Gulcehre, KyungHyun Cho Yoshua Bengio (2014). "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling". In: *NIPS 2014 Deep Learning and Representation Learning Workshop*. arXiv: arXiv:1412.3555.

Michael Phi (2018). *Illustrated Guide to LSTM's and GRU's: A step by step explanation*. URL: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21 (visited on 04/18/2021).

Niklas Donges (2020). *What is transfer learning? Exploring the popular deep learning approach*. URL: https://builtin.com/data-science/transfer-learning (visited on 04/18/2021).

Ord, Keith (1996). "Outliers in statistical data : V. Barnett and T. Lewis, 1994, 3rd edition, (John Wiley and Sons, Chichester), 584 pp., [UK pound]55.00, ISBN 0-471-93094-6". In: *International Journal of Forecasting* 12.1.

Rich Stureborg (2019). *Conv Nets for dummies*. URL: https://towardsdatascience.com/conv-nets-for-dummies-a-bottom-up-approach-c1b754fb14d6 (visited on 04/18/2021).

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). "U-net: Convolutional networks for biomedical image segmentation". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 9351. DOI: 10.1007/978-3-319-24574-4_28.

Thabtah, Fadi et al. (2020). "Data imbalance in classification: Experimental evaluation". In: *Information Sciences* 513. ISSN: 00200255. DOI: 10.1016/j.ins.2019.11.004.

Wen, Tailai and Roy Keyes (2019). *Time Series Anomaly Detection Using Convolutional Neural Networks and Transfer Learning*.