Tomasz CIEPLAK[1], Tomasz RYMARCZYK[2,3], Grzegorz KŁOSOWSKI[1], Michał MAJ[2,3], Damian PLISZCZUK[3], Paweł RYMARCZYK[3]

Lublin University of Technology (1), University of Economics and Innovation in Lublin (2), Research & Development Centre Netrix S.A (3)

doi:10.15199/48.2021.03.28

Methods of process mining and prediction using deep learning

***Abstract**. The first part of the article presents analytical methods to understand how processes (security or business) occur and function over time. The second part presents the concept of a predictive system using deep learning methods that would enable the prediction of subsequent operations or steps that are part of the process under consideration. The article was supplemented with a review of scientific publications related to the content and theoretical foundations were provided. The research was of an applied nature, therefore the considerations are based on the example of analysis and forecasts based on historical data contained in process logs.*

***Streszczenie.** Pierwsza część artykułu przedstawia metody analityczne pozwalające zrozumieć, w jaki sposób procesy (dotyczące bezpieczeństwa lub biznesu) zachodzą i funkcjonują w czasie. W drugiej części przedstawiono koncepcję systemu predykcyjnego wykorzystującego metody głębokiego uczenia, które umożliwiałyby przewidywanie kolejnych operacji lub kroków wchodzących w skład rozważanego procesu. Uzupełnieniem artykułu był przegląd publikacji naukowych pod kątem merytorycznym oraz podano podstawy teoretyczne. Badania miały charakter aplikacyjny, dlatego rozważania opierają się na przykładzie analiz i prognoz opartych na danych historycznych zawartych w logach procesów. (**Metody eksploracji i prognozowania procesów z wykorzystaniem głębokiego uczenia**).*

Keywords: process mining, Petri net, LSTM.
Słowa kluczowe: eksploracja procesów, sieci Petriego, sieci LSTM.

Introduction

The processes, despite their repeatability in time, evaluate and change adapting to the needs of the system in which they are launched. Therefore, it is important to know and understand the changes taking place, and then to create a tool that will be able to predict these changes, in an automated manner. There are many algorithms for solving optimization problems [1-12].

The article has the following structure: the first section provides an overview of other works related to this study; the second section shows the basic terms related to the described subject; the third section presents the techniques related to process mining and an analytical example on a selected set; remaining part is devoted to the implementation of the technique of predicting subsequent events in time that make up the analyzed process; the last section summarizes the topic and contains plans related to future research directions.

Works related to the described research

Process mining is a field that provides methods, tools, and techniques to learn or increase the overall knowledge of a process (usually business [13] but also related to e.g. security [14], [15]). This is done by analyzing the data of occurring events, the data which is collected during the process, and then stored in a specific format. Since process mining is widely used in industry, there is a need to develop various types of tools. Most tools are created for process detection, i.e. creating process models. Another group of tools are those that are used to check compliance, i.e. to verify the extent to which a specific model is a precisely studied process. Process models are also used to improve processes by displaying relevant information, e.g. performance measures. An example of commercial tools widely used in business is *Calonis* [16]. Another group of tools are open systems derived from or used in academic environments. *ProM Framework* is the most widely disseminated tool of this type [17], [18]. The third group of software are libraries that enable attaching opportunities related to process mining in the built systems. The experiment uses the *PM4Py* library for the Python language [19]. The research was inspired by the article [20] in which the authors presented the assumptions of predictive monitoring of business processes as well as on previous works of authors [21].

Concepts and theoretical foundations

This part of the article presents the definitions and concepts used in subsequent sections of the document.

Analysis of event logs. Petri nets

The Petri net is a bipartite directed graph whose vertices belong to two sets: places and transitions. Arcs can only connect two types of elements: places and transitions. The state of the network is determined by markers (tokens) located in places and moving as a result of executing (firing) transitions. The Petri net is a convenient tool for modeling related concurrent phenomena and is often used in work on analyzing the operation of digital machines and their software. The basic structure of the network is static but, according to the burning principle, tokens can flow through the network. The state of the Petri network depends on the distribution of tokens in different places and is referred to as its tagging. There is only one character in the start tag - the start is the only place tagged. However, multipart sets are not only used to represent signs; later, you should use multiple assemblies to model log events where the same trace may appear multiple times.

Therefore, the basic concepts that are used in Petri nets are marking, tokens, arches, places, and transitions. A Petri net is defined as a tuple $PN = (P, T, F, W, M_0)$, wherein: P is a finite set of places; T is a finite set of transitions; $F \subset (P \times T) \cup (T \times P)$ is a finite set of arcs; $W: F \rightarrow N$ is a function that assigns weights to arches; $M_0: P \rightarrow N \cup \{0\}$ is the initial marking. $N = (P, T, F, W)$ often denotes a network without a specific initial marking. Through (N, M_0) a network with a fixed initial marking [22].

In the special case, when all the distribution of activation times are pointwise, a Petri net with a deterministic firing time of transitions is obtained, often used to model the operation of digital machines. There may be competition between the transitions for the marker in the Petri nets. Launching one of the passages takes the marker from the spot and you cannot fire the other pass. It is therefore necessary to define a criterion for selecting one of the roads. The easiest way is to determine the probabilities P_i of activation of individual passes and the distribution Φ_i determining the random distribution of firing time of each of the passes if selected. Then the probability of firing the L_i transition over time t is determined by the function $P_i\Phi_i(t)$. The dynamic behavior of a Petri net marked in this way is defined by the so-called firing rule [23]. The transition as well as the rule are enabled if each of its inputs contains a token. The merged transition thus consumes one token from each entry slot and produces one token for each exit slot.

A Petri net is a workflow that meets three conditions: there is a unique source site (part of the pre-tag), there is a unique take-up site (part of the final tag), and each site and transition is in the path from the source site to the dumpsite.

The process shown in Fig. 1 shows the Petri net as a model of the custom furniture manufacturing process. This type of analysis allows for a parametric improvement of the existing system employing a Petri net.

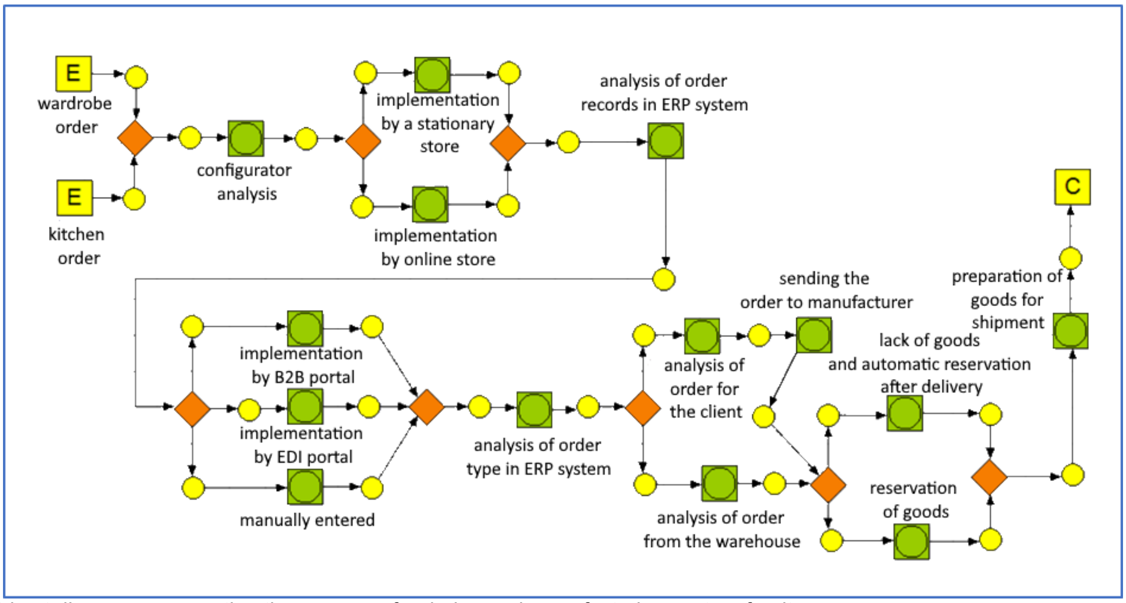

Fig. 1. Petri net diagram representing the process of ordering and manufacturing custom furniture.

Event log data

The source of information for the Petri nets is data collected in event logs. The trace is a finite non-empty sequence of events so that each event occurs only once and the time does not decrease. The event log is a set of traces, thanks to which each event appears maximum once in the entire log. When building forecasts, the data from the classic XES format is transformed into the transition sequences format.

Table 1. List of conversion transitions to the abbreviations used in the log file.

Transitions name	Abbreviation in the log file
wardrobe order	WO
kitchen order	KO
configurator analysis	CA
implementation by a stationary store	ISS
implementation by online store	IOS
analysis of order records in the ERP system	AORERP
implementation by B2B portal	IB2B
implementation by EDI portal	IEDI
manually entered	ME
analysis of the order type in the ERP system	AOT
analysis of order for the client	AOC
analysis of order from the warehouse	AOW
sending the order to the manufacturer	SOM
lack of goods and automatic reservation after delivery	LGAR
reservation of goods	RG
preparation of goods for shipment	PGS

Table 2. Example of the process statistics.

From	Wait time [h]	cycle time [h]	work time [h]	cost [$]
kitchen_order	0	151.43	151.43	126.31
kitchen_order	0	133.05	133.05	82.74
wardrobe_order	2.07	138.01	135.94	30.87
wardrobe_order	14.91	177.86	162.95	106.38
wardrobe_order	10.02	184.37	174.35	99.59
kitchen_order	15.58	149.66	134.09	30.5
wardrobe_order	15.76	162.76	147	79.39
kitchen_order	19.76	139.09	119.33	27.55
wardrobe_order	0	191.21	191.21	129.85
kitchen_order	18.84	153.35	134.51	57.15
kitchen_order	15.6	223.98	208.39	79.95
kitchen_order	20.46	134.22	113.76	76.77
kitchen_order	4.12	174.76	170.65	37.81
wardrobe_order	42.78	173.84	131.06	29.89
kitchen_order	71.04	223.92	152.88	57.43
kitchen_order	37.57	227.13	189.55	127.74

Examples of transition sequences logs of registered processes saved in the SQL table are presented in Fig. 2.

Results Messages

	log
1	AORERP IB2B AOT AOC SOM RG
2	KO CA IOS AORERP ME AOT
3	RG PGS
4	WO CA IOS AORERP IEDL AOT AOC SOM
5	AOW LGAR

Fig. 2. Example of the transition sequence format.

Due to the complexity of the processes as well as the convenience of processing and facilitating the subsequent analysis of data resulting from the logs, the names of the processes presented using the Petri nets (see Fig. 1) have their abbreviations. These abbreviations are presented in the Table 1.

The following table (see Table 2) presents the statistics recorded for individual complete order fulfillment processes, both for cabinets and kitchen furniture. Thanks to this, we can also analyze the time of execution of a specific order from its inception, through the production of the product to its shipment to the target customer.

Timeseries prediction with LSTM

LSTM (Long Short-Term Memory) is used that is capable of learning long-term relationships. Therefore models trained with LSTM apply to jobs such as speech recognition, anomaly detection, long-term relations analysis, traffic analysis. Remembering information for a long time is practically their default behavior, not something they are trying to learn. LSTM is a special case of recursive neural networks, and just like them, it has a chain-like structure, but the repeating module has a different structure. Instead of a single layer of a neural network, there are four layers that interact with each other in a very special way.

Taking into consideration (1), the LSTM model checks h_{t-1} and x_t and derives a number from 0 to 1 for each number in the c_{t-1} cell state. A value of 1 means “keep it completely”, while a value of 0 means “forget it completely”. Then, one needs to specify what information should be stored in the cell state. Initially, the sigmoidal layer, called the input layer, decides which values will be updated (2).

(1) $$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$
(2) $$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
(3) $$g_t = tanh(W_g \cdot [h_{t-1}, x_t] + b_g)$$

Next, the tanh layer produces as its output a vector of new candidate values, denoted as g_t, which will be added to the state (3). Then, these two steps are combined to create an update of the original state of the c_{t-1} cell to the new state of the c_t cell. In (4), it is described that the old state c_{t-1} should be multiplied by the ratio of forgetful states f_t, then the multiplication result is added to the product of the vector values of the candidates g_t to scale each state value for updating [24].

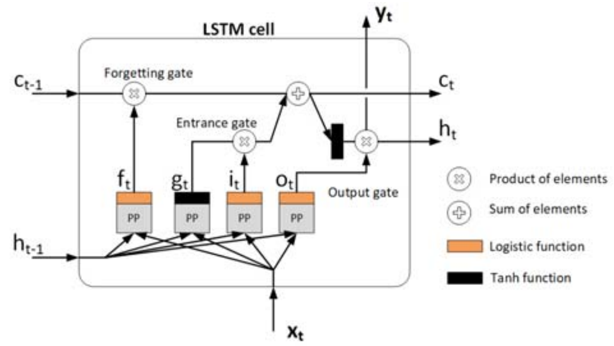

Fig. 3. Structure of LSTM cell.

(4) $$\boldsymbol{C_t = f_t \times C_{t-1} + i_t \times g_t}$$

The final step is to determine the output values by calculation based on the cell states that are filtered. At this stage, the sigmoidal layer decides about the states of the cells to go out (5), and then the filtered states of the cells pass through the tanh function to convert the values to values in the range from -1 to 1.

(5) $$\boldsymbol{O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)}$$
(6) $$\boldsymbol{h_t = O_t \times \tanh(C_t)}$$

Finally, the output (equation 6) is the value of the ratio of the function tanh, which is multiplied by the output of the sigmoid gate.

Techniques related to the analytical process

In this chapter, techniques related to process mining and an analytical example on a selected set will be discussed.

Data set description

Table 3 summarizes the data set, which contains 27,000 events.

Table 3. Dataset description

	Data Type	Count	Uni. Val.	Mis. Val.
concept:name	**object**	**27000**	**16**	**0**
lifecycle:transition	**object**	**27000**	**1**	**0**
time:timestamp	**datetime64[ns]**	**27000**	**27000**	**0**

Process events prediction with LSTM

To predict subsequent sequences of the order process, the previously described LSTM networks were used. To increase their accuracy and improve the learning process, the recorded sequences of transitions have been divided into three-element subsequences with a one-step element shift between successive sequences (moving window). The text data has of course been converted to numeric values for model learning. Based on the received name of a given element of the input process, the model is to predict three consecutive steps of this process. Keras Framework was used to build the test environment. The figure shows the learning process of successive layers of the model.

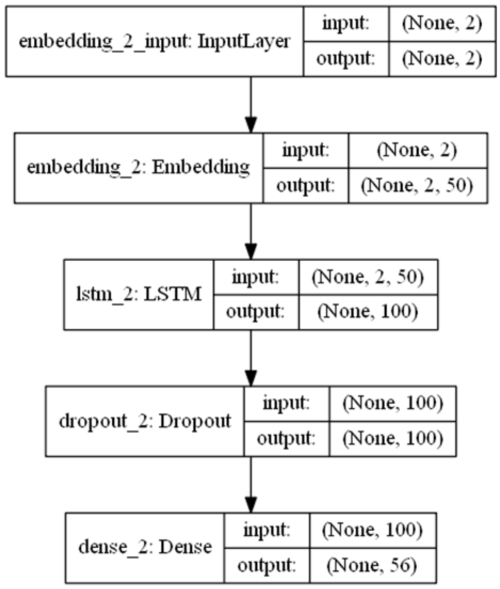

Fig. 4. Layers diagram of the LSTM network.

The table below presents a summary of the built model.

Table 4. Summary of the LSTM network.

Layer (type)	Output Shape	Param #
embedding_2 (Embedding)	(None, 2, 50)	2800
lstm_2 (LSTM)	(None, 100)	60400
dropout_2 (Dropout)	(None, 100)	0
dense_2 (Dense)	(None, 56)	5656

Total params:	68,856
Trainable params:	68,856
Non-trainable params:	0

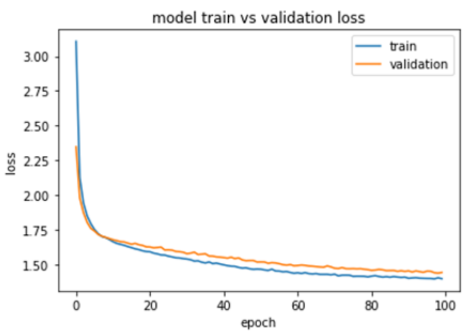

Fig. 5. Learning and validation curve of the trained network.

Results

The learning and validation curve is shown in Fig. 5. This graph shows a good fit of the model to the data. The popular MEA measure was used as the metric and 'categorical_crossentropy' was used as a loss function. The following hyperparameters were found using the grid search ethod: 'activation': 'softmax', 'optimizer': 'adam'.

Application of a trained model in the IT system

The visualization of the prediction results is carried out by means of a microservice. The task of the microservice is to analyze the processes with the use of the LSTM network. On the basis of the learned model, a sequence of the next three most frequently used stages in the studied process is generated. Inside the microservice, the learned LSTM network is based on the Keras library and the Tensorflow environment.

Based on the trained model, an example prediction obtained by using the AOC task as initiator is shown in the next figure.

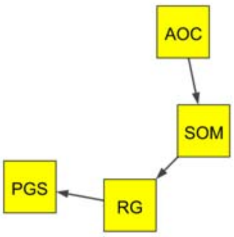

Fig. 6. Visualization of prediction of subsequent stages of the process using LSTM.

Summary

The use of deep learning methods together with the classic methods of business process modeling seems to be a good way to optimization. The presented result of the experiment is the foundation for an independent system used for forecasting and optimization of business processes. Therefore, it is worth considering process modeling using Petri nets and various predictive models to optimize many business processes.

Acknowledgment

This research was supported under the program RPLU.01.02.00-06-0031/16-00 "Stworzenie inteligentnego systemu analizy danych wieloźródłowych służącego do monitorowania zachowań konsumenckich oraz urządzenia do przesyłania danych z systemów POS"

***Authors**: Tomasz Cieplak, D.Eng. Lublin University of Technology, e-mail: t.cieplak@pollub.pl; Tomasz Rymarczyk, DSc Ph.D. Eng., University of Economics and Innovation, Projektowa 4, Lublin, Poland/ Research & Development Centre Netrix S.A. e-mail: tomasz@rymarczyk.com; Grzegorz Kłosowski, Ph.D. Eng. Lublin University of Technology, e-mail: g.klosowski@pollub.pl; Michał Maj, M.Sc. Eng., University of Economics and Innovation, e-mail: michal.maj@wsei.lublin.pl; Damian Pliszczuk, M.Sc. Eng., Research & Development Centre Netrix S.A. e-mail: damian.pliszczuk@netrix.com.pl*

REFERENCES

[1] Romanowski, A. Big Data-Driven Contextual Processing Methods for Electrical Capacitance Tomography. IEEE Trans. Ind. Informatics, 15 (2019), 1609–1618
[2] Rymarczyk T., Characterization of the shape of unknown objects by inverse numerical methods, Przegląd Elektrotechniczny, 88 (2012), No. 7b, 138-140
[3] Rymarczyk T., Kłosowski G., Tchórzewski P., Cieplak T., Kozłowski E.: Area monitoring using the ERT method with multisensor electrodes, Przegląd Elektrotechniczny, 95 (2019), No. 1, 153-156
[4] Rymarczyk T., Nita P., Vejar A., Woś M., Stefaniak B., Adamkiewicz P.: Wearable mobile measuring device based on electrical tomography, Przegląd Elektrotechniczny, 95 (219), No. 4, 211-214
[5] Kłosowski G., Rymarczyk T., Kania K., Świć A., Cieplak T., Maintenance of industrial reactors based on deep learning driven ultrasound tomography, Eksploatacja i Niezawodnosc – Maintenance and Reliability; 22 (2020), No 1, 138–147
[6] Kłosowski G., Rymarczyk T., Wójcik D., Skowron S., Adamkiewicz P., The Use of Time-Frequency Moments as Inputs of LSTM Network for ECG Signal Classification, Electronics, 9 (2020), No. 9, 1452
[7] Kłosowski G., Rymarczyk T., Cieplak T., Niderla K., Skowron Ł., Quality Assessment of the Neural Algorithms on the Example of EIT-UST Hybrid Tomography, Sensors, 20 (2020), No. 11, 3324
[8] Sekulska-Nalewajko, J., Gocławski, J., Korzeniewska, E., A method for the assessment of textile pilling tendency using optical coherence tomography, Sensors (Switzerland), 20 (2020), No.13, 1-19, 3687
[9] Pawłowski, S., Plewako, J., Korzeniewska, E., Field modeling the impact of cracks on the electroconductivity of thin-film textronic structures, Electronics (Switzerland), 9 (2020), No.3, 402
[10] Fraczyk, A.; Kucharski, J. Surface temperature control of a rotating cylinder heated by moving inductors. Appl. Therm. Eng., 125 (2017), 767–779
[11] Kozłowski E., Mazurkiewicz D., Żabiński T., Prucnal S., Sęp J., Machining sensor data management for operation-level predictive model. Expert Systems with Applications 159 (2020), 1-22
[12] Véjar A., Charpentier P., Generation of an adaptive simulation driven by product trajectories, J Intell Manuf, 23 (2012), No. 6, 2667–2679
[13] Van Der Aalst W. M. P. et al., Business process mining: An industrial application, Inf. Syst., 32 (2007), No. 5, 713–732,
[14] Bernardi S., Alastuey R. P., and Trillo-Lado R., Using process mining and model-driven engineering to enhance security of web information systems, in 2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), 2017, 160–166.
[15] Mishra V. P., Dsouza J., and Elizabeth L., Analysis and comparison of process mining algorithms with application of process mining in intrusion detection system, in 2018 7th International Conference on Reliability, Infocom Technologies and Optimization, 2018, 613–617
[16] Vei F. t, Geyer-Klingeberg J., Madrzak J., Haug M., Thomson J., The Proactive Insights Engine: Process Mining meets Machine Learning and Artificial Intelligence. in BPM (Demos), 2017
[17] Kalenkova A., De Leoni M., and. M. P. van der Aalst W. M. P., Discovering, Analyzing and Enhancing BPMN Models Using ProM. in BPM (Demos), 2014, 36.
[18] Verbeek H. M. W., Buijs J., Van Dongen B. F., and van der Aalst W. M. P., Prom 6: The process mining toolkit, Proc. BPM Demonstr. Track, 615 (2010), 34–39, 2010
[19] Berti A., van Zelst S. J., and van der Aalst W., Process mining for Python (PM4Py): bridging the gap between process-and data science, arXiv Prepr. arXiv1905.06169, 2019.
[20] Tax N., Verenich I., Rosa M. La, and Dumas M., Predictive Business Process Monitoring with LSTM Neural Networks, 2017
[21] Kłosowski G., Gola A., and Świć A., Human Resource Selection for Manufacturing System Using Petri Nets, in Applied Mechanics and Materials, 791 (2015), 132–140
[22] Huang Y.-S. and Pan Y.-L., A Computationally Improved Control Policy for FMS Using Crucial Marking/Transition-Separation Instances, in Formal Methods in Manufacturing Systems: Recent Advances, IGI Global, 2013, 61–82
[23] .Halder A. and Venkateswarlu A., A study of Petri nets modeling Analysis and simulation, Dep. Aerosp. Eng. Indian Inst. Technol. Kharagpur, India, 2006
[24] .Zhang W. et al., LSTM-based analysis of industrial IoT equipment, IEEE Access, 6 (2018), 23551–23560