DATA MINING APPLIED IN FOOD TRADE NETWORK

Alessandro Massaro, Giovanni Dipierro, Annamaria Saponaro and Angelo Galiano

Dyrecta Lab, IT Research Laboratory, via Vescovo Simplicio, 45, 70014 Conversano (BA), Italy

ABSTRACT

The proposed study deals with the design and the development of a Decision Support System (DSS) platform suitable for the global distribution system (GDS). Precisely, the prototype platform combines artificial intelligence and data mining algorithms to process data collected into a Cassandra Big Data system. In the first part of the paper platform architectures together with all the adopted frameworks including Key Performance Indicators (KPIs) definitions and risk mapping design have been discussed. In the second part data mining algorithms have been applied in order to predict main KPIs. The adopted artificial neural networks architectures are Long Short-Term Memory (LSTM), standard Recurrent Neural Network (RNN) and Gated Recurrent Units (GRU). A dataset with KPIs has been generated in order to test the algorithms. All performed algorithms show a good matching with the generated dataset, thus proving to be the correct approach to predict KPIs. The best performances in terms of Accuracy and Loss are reached by using the standard RNN. The proposed platform represents a solution to increase the Knowledge Base (KB) for a strategic marketing and advanced business intelligence operations.

KEYWORDS

KPI, Big Data, Data Mining, Artificial Neural Networks, Cassandra, Strategic marketing, Business Intelligence.

1. INTRODUCTION AND PRELIMINARY SPECIFICATIONS OF THE RESEARCH PROJECT

The gain of the knowledge base -KB- covers a crucial role in scientific research projects involving the information system upgrade and Enterprise Resources Planning integration [1]-[5]. In the light of this, data mining algorithms can improve the KB by means of Decisional Support Systems (DSS) with the aim to engineer production processes [6]-[9]. In the global distribution system (GDS), DSS plays a fundamental rule for strategic marketing [10]-[14]. By investigating these main topics in the context of an industry research project oriented on the creation of a GDS services platform. the architecture shown in Fig. 1 has been proposed. Food trade network is constituted by many actors working in the whole supply chain. Data mining could support the business intelligence of the activities to perform in this sector by defining properly structured Key Performance Indicators -KPIs- of Small and medium-sized enterprises (SMEs), suppliers and agents as main actors of the supply chain. Furthermore, important factors addressing strategic marketing for the food trade network is the dynamic formulation of price lists. These

DOI: 10.5121/ijaia.2020.11202

specifications provide the scenario shown in Fig. 1 where data and information about prices, market competitors and other data could be processed by a data mining engine providing as outputs the KPIs to optimize the marketing processes. The goal of the project and of the proposed research is the formulation of innovative algorithms gaining the KB and predicting KPI thus making the platform innovative for the specific case of study. According with the project goal, some works proposed in literature the formulation of KPIs in logistics services [15] and in whole supply chain activities [16]-[18]. In particular in [16] are distinguished the primary activities to support activities by defining a different KPI metrics. In [17] is proposed a structured framework for creating and evaluating supply chain performance indicators, and in [18] have been proposed balanced scorecard as customers, internal processes, innovations, and finance indicators. In the proposed paper are analysed some innovative methodologies to estimate KPI predicting values and defining a framework useful for industries operating in food supply chain.

Based on these premises, the work hereby presented is structured as follows:

(i) Design of the information system platform by defining all KPIs supporting the DSS;
(ii) Design of the risk model as a supplementary tool for DSS;
(iii) Testing of the platform by executing the artificial neural networks predicting KPIs;
(iv) Conclusions.

2. PLATFORM DESIGN

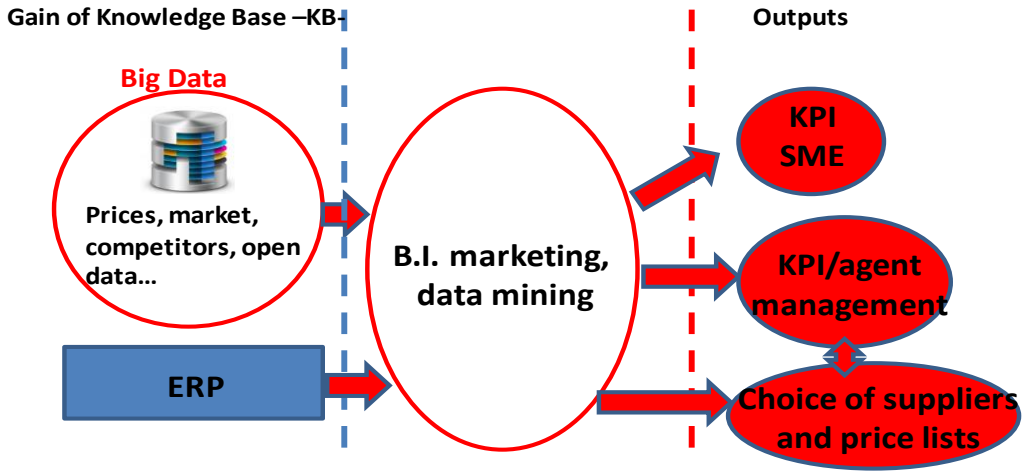

Figure 1. Prototype system architecture of the proposed platform.

The research project involves the implementation of the following main project modules:

1. **Digitalization/information integration module associated with process engineering** (speeding up the process activation and KPIs evaluation measuring industry performances [15]-[18]). This module includes the development of functional modules for the creation of a new KB. It is oriented to the digitalization of information and on the management of basic functions and activities of the company. The implementation of this module will have the purpose of integrating multiple information systems and multiple

types of information (structured and unstructured). This integration will allow to speed up the management of the company's activities and to define and optimize the new KPIs with the help of an innovative data mining engine. The KPIs management will provide indicators useful for the achieving of economic benefits and for the definition of profit margins.

2. **Process mapping module**. The current company processes ("AS IS" processes) are mapped in order to direct research to well-defined application processes. Based on this analysis, new "TO BE" processes are formulated based on the analysis of the outputs of the applied data mining algorithms.

3. **Big Data module** (collector of the new digitalized KB). A Big Data [19] system is designed and developed in order to provide a massive amount of data as input to the data mining algorithms. Other data sources such as open data, market data, social data, etc. could be integrated into this system in order to create predictive models with the lowest possible predictive error rate.

4. **Module of innovative data mining/artificial intelligence algorithms**. This module provides the implementation of new flow charts and models of data mining algorithms, whose innovation will be guaranteed by the analysis of the updated state of the art and by the design on a scientific/systemic basis, starting from the concepts discussed in this analysis of pre-feasibility.

The platform provides new outputs for the enriched KB, allowing at the same time to analyse large amounts of data. The data processing to perform will support the strategic marketing, and the integrated logistics. The structured data and non-fragmentary information system will perform intelligent price analyses and subsequently obtain greater bargaining power. The platform will provide different performance dashboards suitable for a systemic performance assessment of the food trade network.

The preliminary architecture of Fig. 1 is further detailed in Fig. 2, where are listed the main technologies involved such as Vertica Big Data and Django framework. Besides the main system actors such as the system administrator (Admin) and the user (Operatore) are specified, input data are then processed by means of the DSS engine returning as outputs KPI results.

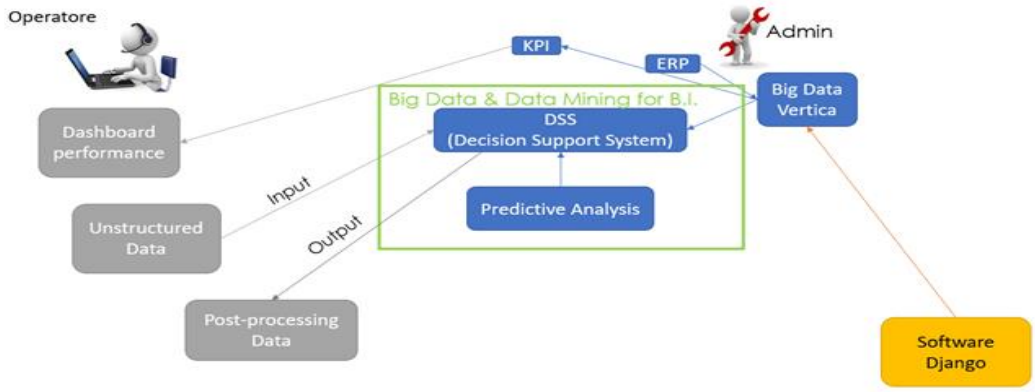

Figure 2. System architecture of the proposed platform.

The diagram in Fig. 3 represents modules that are integrated in the system architecture, schematically organized by packages. In particular, the technologies used for each package are the following:

• Data Mining engine constructed by means of DSS and predictive analysis (artificial intelligence algorithms);

• Admin dashboard and operator dashboard: Django framework based on Python language;
• Vertica Big Data system;
• Different ERP software providing data to process;
• Other external databases linked to the platform.

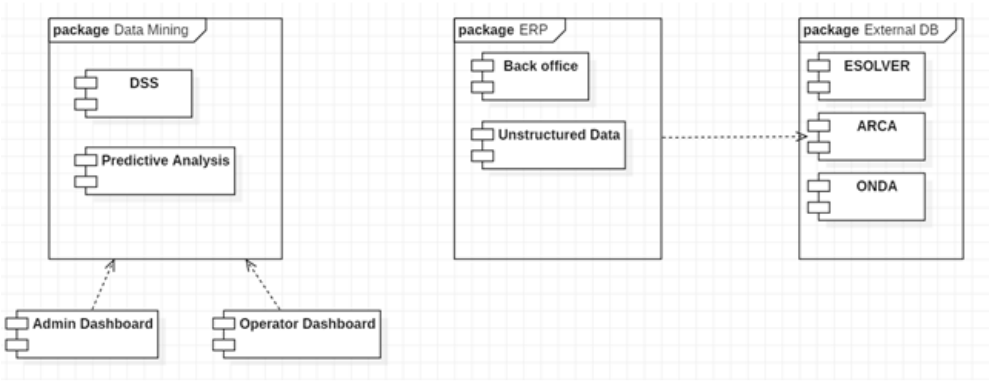

Figure 3. Platform system architecture structured with packages.

Figure 4 illustrates the Unified Modeling Language (UML) of functional scheme linking all packages and system actors.

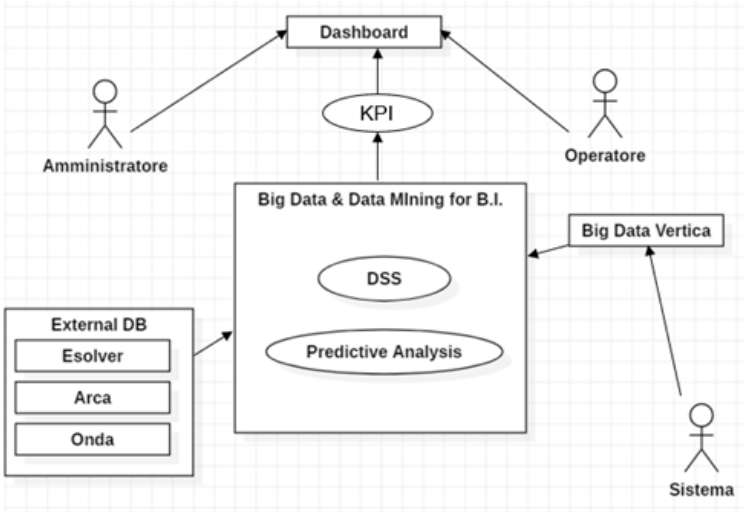

Figure 4. UML functional scheme of the proposed architecture involving different actors and system packages.

KPIs are provided by the engine consisting of Big Data and data mining algorithms. Among those KPIs which are useful for business intelligence can be found:

- the financial status of existing customers;
- the number of new acquired customers;
- the abandonment rate of customers;
- customer segmentation by profitability or demography;
- the waiting time for customer orders;
- the length of the stock outs.
- the average cost of human resources per hour worked;
- the company's ability to manage the workers.

In particular, the following indicators have been identified:

- performance of workers: the attention is focused on the ability of workers directly involved in production process to transform their working hours into production volumes (operators engaged in the production units in all processing phases);
- indirect incidence: the attention is focused on worked hours of human resources not directly involved in the production process but equally necessary to plant operations (administrative, responsible for planning and controlling production, etc.);
- incidence of overtime: the attention is focused on the percentage of hours worked that occur overtime (the productivity of human resources is a first level “basic” indicator).

Additional important indicators include human resources (HR) indicators, such as:

- Human resources management;
- Productivity of human resources;
- Voluntary staff turnover rate;
- Product Compliance Management;
- Number of complaints;
- Number of restraints as ability to manage customers;
- Number of detected non-conformities as an index of the correct functioning of business processes;
- Number of detected and unresolved non-conformities as representative of business efficiency;

Table 1 shows other KPIs dealing with production activities able to improve the DSS engine.

Table 1. List of KPI in production activities.

KPI Topic	KPI Description
Productivity	-Items or transactions processed by employee; -Orders shipped per hour by employee; -Increase or decrease in inventory by item; -Inventory turnover ratios; -Comparison between automated, semi-automated, manual processes; -Time processing reduction;
Quality	- Cost of quality inspection;

	- Number of customer complaints -Warrantly claims; - Returns and allowances; - Good units produced; - Product availability; -Sales forecasting.
Transportation	- Carriers; -Vehicle costs; - Fixed operational cost; - Transport costs; - Overhead costs; -Shippers; - Inventory costs;
Processing	- Time to process an order; - Time to process shipping; - Time to process billing.
Accounting	- Number of payments at terms time; - Numbers of bills at time of shipment; - Invoices/payments.
Facilities/Capacity	- Human resources number; - Number of quality inspection numbers; - Smoothness of workflow; - Ability to schedule; - Responsibility for work areas; - Number of changes;

3. RISK MAP DESIGN

The KPIs are important indicator for risk management. From the analytical point of view, "risk" (R) is defined as the combination of a potential hazard, quantified in terms of probability of occurrence of the event (P), event exposure (E), and related vulnerability (V) [20]:

$$R = P * E * V \quad (1)$$

Following a classic approach, risk is represented by means of risk maps, a powerful visualization tool that allow to assign a priority order of corrective actions by examining the positioning of the highlighted risk scenarios.

Events characterized by a high probability of occurrence and limited damage size can be classified as having the same level of risk of other events with even lower probability but much greater damage dimensions. In this way, it is possible to define acceptable risk levels, which in turn allow the identification of tolerable combinations of event probability / damage size.
It has to be highlighted that this identification provides risk managers with a tool to comprehensively understand risks as a whole of events and conditions causing them. Moreover, risk classification allows the company to decide which entity within a company is responsible for managing a particular risk category.

Essentially, in the specific area of large-scale distribution, the main areas potentially at risk are:

- Business Continuity Management;
- Credit and Deposits;
- Job security;
- Supply Chain;
- Contracts and liability towards third parties (customers and suppliers);
- Loss Prevention.

Risk assessment is essential for the subsequent classification in different categories. For each identified risk area, it is possible to quantify the probability of occurrence of the event and the impact associated with that event. The probability of occurrence is related to the probability of a certain event happening. For each risk area four levels of probability were considered, ranging from "1" (improbable) to "4" (very probable). Similarly, the impact refers to the consequences associated with the occurrence of the event, and is quantified by defining 4 levels, ranging from "1" (negligible) to "4" (catastrophic). Subsequently, a class-by-class multiplication approach is applied for risk determination: class 1 of the probability is multiplied by class 1 of the impact giving a value equal to 1, class 1 of the probability is multiplied by class 2 of the impact returning as a result 2, and so on. The final values are presented in matrix form, where the values between 1 and 2 (in yellow) indicate a "low" level of risk, the values between 3 and 6 (in light orange) indicate an "average" level of risk, the values between 8 and 9 (in dark orange) indicate a "high" level of risk, and finally the values between 12 and 16 (in red) indicate a "very high" level of risk [20].

In this way it is possible to quantify the total risk relating to each exposed area by means of a matrix where 4 different levels are displayed (Figure 5).

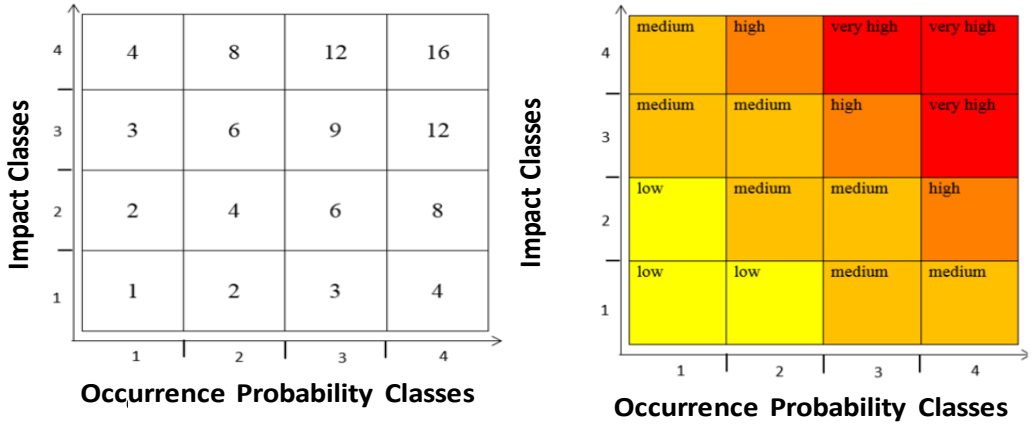

Figure 5. Multiplicative approach for calculating the level of risk. First, each occurrence probability class is multiplied by each impact class (left); subsequently the values are classified as "low", "medium", "high" and "very high" (right).

Once risk levels have been quantified, the next step is represented by defining an appropriate strategy for managing each level. Specifically, this strategy concerns the definition of the risk acceptability threshold and the identification of the following levels:

- the acceptable level of risk, for which no control measures need to be implemented;

- the level of monitoring, which includes those risks that require constant monitoring;
- the exclusion level, which refers to the risks that exceed the acceptability threshold.

Hence, starting from the definition of these levels, it is possible to define specific activities aimed to risk prevention as well as mitigation of individual risks.

With particular reference to risk management applied to the supplier chain, it has to be considered that even structural elements of the supply chain can be real risk drivers. In particular, among all possible structural elements, the following ones can be identified as risk factors:

- poor reliability of suppliers and/or key customers, in financial or competitive terms;
- lack of information technology integration or visibility between chain partners;
- limited number of suppliers and/or key customers;
- high stock;
- very long chain lead time, due to product characteristics and related processes.

4. DSS ALGORITHMS RESULTS

The artificial intelligence (AI) algorithms supporting the DSS have been implemented by means of recurrent neural networks. Different architecture of neural networks such as Long Short Term Memory (LSTM) neural networks [13],[21], standard Recurrent Neural Network (RNN) [22],[23] and Gated Recurrent Units (GRU) [24] have been tested in terms of accuracy. The comparison of results of the LSTM network with the different RNN and GRU architecture has been implemented through a code written in Python language (see appendix A). A comparison of the three mentioned approaches is reported in Fig. 6; as it can be seen there is a good matching with the main dataset trend, thus confirming that all the neural networks can be adopted for DSS.

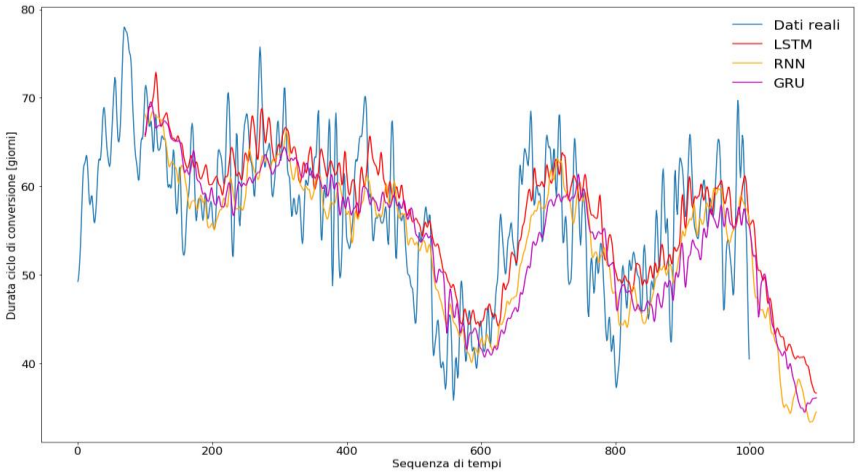

Figure 6. Comparison between LSTM (red line), RNN (yellow line) and GRU (purple line) results with dataset trend (blue line). The plot is related to activity duration expressed in days ("Durata ciclo di conversione") versus time ("Sequenza di tempi").

In order to evaluate which model best approximates the dataset trend, we estimate the accuracy of the model, by calculating the cosine of the angle between the vectors that represent the input

dataset and the prediction. Note that in the three cases, the models converge rapidly to a cosine value of 1, indicating a good aggrement of the predicted model with the data (see Fig. 7). Furthermore, the inverse cosine function (arc cosine) has been applied to obtain the typical loss function decreasing to 0 (see Fig. 8).

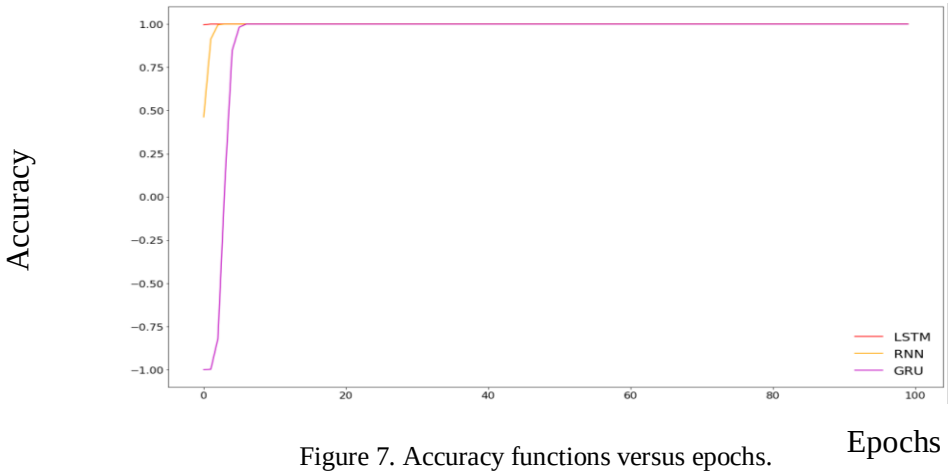

Figure 7. Accuracy functions versus epochs.

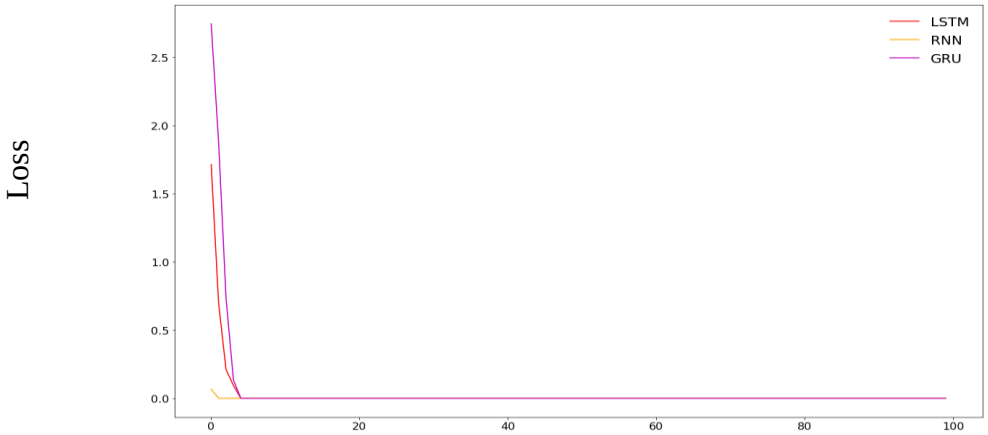

Figure 8. Loss function versus epochs.

The test carried out show that RNN represents the best tool for predicting values. With regard to the case study, KPIs of a single sales agent or the entire sales team represent a set of values that can be of different nature and correlated with each other in a non-trivial way.

Three main characteristics of KPIs concerning sales agent have been selected: the duration of the conversion cycle (DCC), the percentage of new customers (PNC), and the acquired references (RA). Dataset adopted to check the models are generated by a linear combination of sines and cosines with amplitude and random period. The analytical functions adopted for the calculus are:

$$DCC(t) = random(1,20) \cdot \sin(random(0.005,01) \cdot t) + \text{random}\ (1,10)) \cdot sin\big(random(0.02,0.04)\big) \cdot t + random(40,70) \tag{2}$$

$$PNC(t) = random(1,20) \cdot cos(random(0.005,01) \cdot t) + \text{random}\ (1,10)) \cdot \cos\big(random(0.02,0.04)\big) \cdot t + random(40,70) \tag{3}$$

$$RA(t) = random(1,20) \cdot \sin(random(0.005,01) \cdot t) + \text{random}\ (1,10)) \cdot \cos\big(random(0.02,0.04)\big) \cdot t + random(40,70) \tag{4}$$

where random(a,b) indicates the generator of random numbers in the range [a,b].

Due to the randomization of the amplitude and period of these sinusoids, the correlation between the different quantities will not be very significant. In Fig. 9 the KPI comparison between generated data (DCC, PNC, RA) and predicted ones (DCC predetti, PNC predetti, RA predetti) is illustrated; in particular, Gaussian smoothing has been applied in order to better highlight each plot trend. Comparisons between predicted results and generated datasets are better illustrated in Fig. 10, 11 and 12 by observing a good convergence of the three adopted algorithms. Only for the PNC trend it is possible to observe a better convergence of the RNN network.

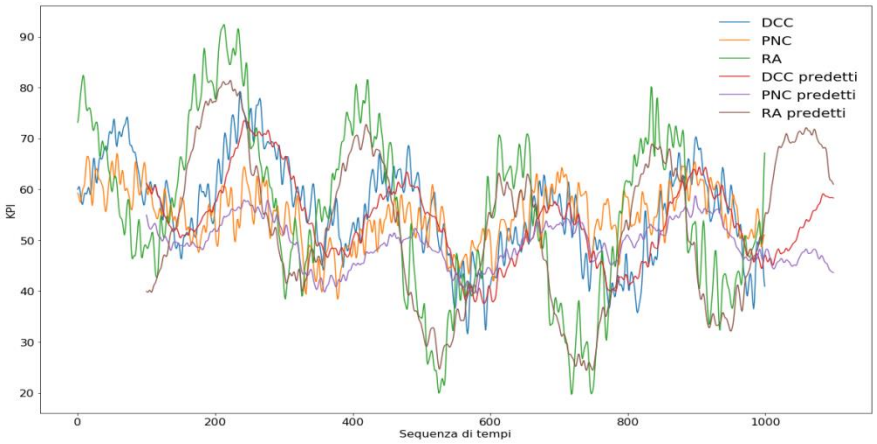

Figure 9. Comparison of KPI trends (data generators) and predicted KPI versus time.

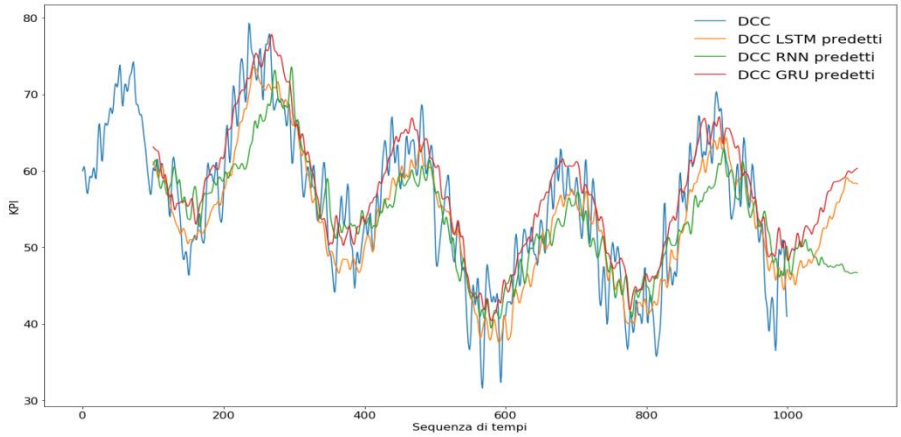

Figure 10. Comparison of DCC trend and predicted KPI versus time (LSTM, RNN, GRU).

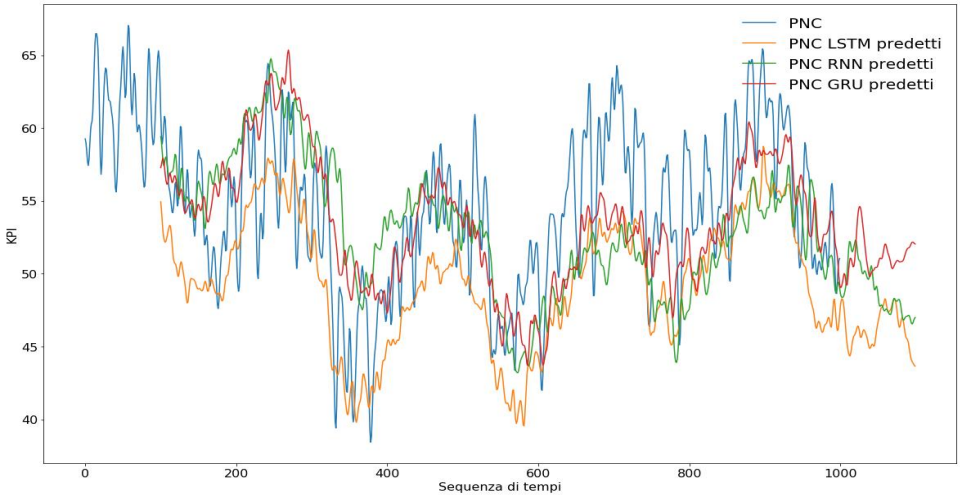

Figure 11. Comparison of PNC trend and predicted KPI versus time (LSTM, RNN, GRU).

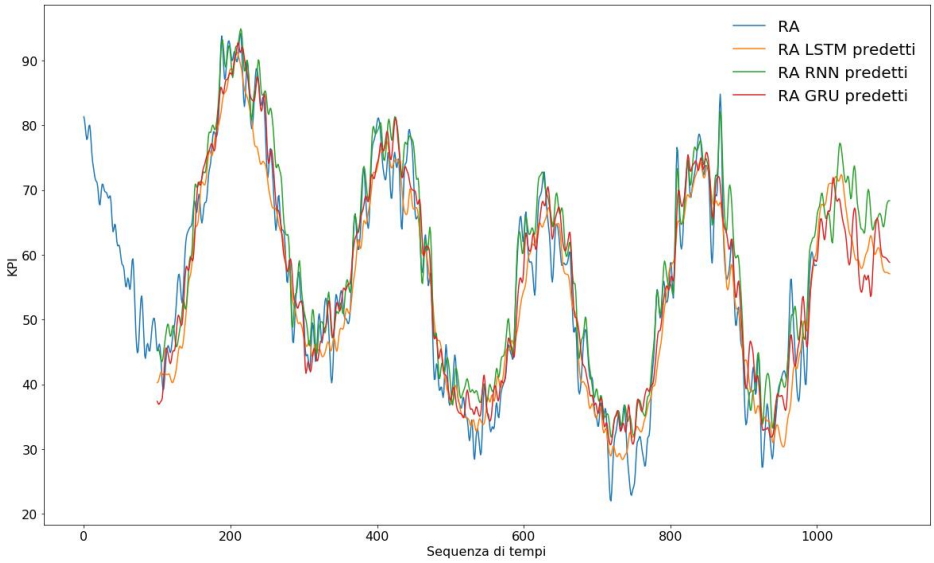

Figure 12. Comparison of RA trend and predicted KPI versus time (LSTM, RNN, GRU).

The main scripts of the implemented neural network algorithms are listed in Appendix A. The dataset has been stored into a Cassandra NoSQL Big Data system. The Big Data table configuration setting has been discussed in Appendix B.

5. CONCLUSION

The paper proposed a platform and then discussed the results of a research industry project related to the design and the development of a DSS platform aimed at predicting KPIs of a company working in GDS. The performed design involves system architecture, general and specific KPIs specifications, together with a risk mapping approach and dataset generator allowing the testing of artificial intelligence algorithms. The proposed platform implements LSTM, RNN and GRU neural network algorithms which have been tested in terms of accuracy, predicting KPIs formulated for the specific case of study such as conversion cycle (DCC), percentage of new customers (PNC), and acquired references (RA). In order to verify the models has been executed a dataset generator for each kind of KPI. Results show the best performances of KPIs prediction is reached by the RNN approach. The implemented Cassandra Big Data technology will be adopted also to predict other relevant KPIs in order to enhance risk evaluation by processing multiple variables. The proposed DSS platform represents a systematic approach to increase the knowledge base addressing industry activities on advanced business intelligence strategies.

ACKNOWLEDGEMENTS

The work has been developed in the frameworks of the Italian projects (“Algoritmi innovativi di data mining applicati a dati di gestione delle attività connesse alla rete di commercio di beni alimentari mediante il supporto di tecnologie Big Data: ‘BIG DATA MINING FOOD STRATEGIC MARKETING’" ["Innovative data mining algorithms applied to management data

of activities connected to the food trade network through the support of Big Data technologies: 'BIG DATA MINING FOOD STRATEGIC MARKETING'].

REFERENCES

[1] Massaro, A., Maritati, V., Galiano, A., Birardi, & Pellicani, L. (2018) "ESB platform integrating KNIME data mining tool oriented on Industry 4.0 based on artificial neural network predictive maintenance," International Journal of Artificial Intelligence and Applications (IJAIA), Vol. 9, No. 3, pp 1-17.

[2] Massaro, A., Lisco, P., Lombardi, A., Galiano A., & Savino N. (2019) "A case study of research improvements in an service industry upgrading the knowledge base of the information system and the process management: data flow automation, association rules and data mining," *International Journal of Artificial Intelligence and Applications (IJAIA)*, Vol. 10, No. 1, pp 25-46.

[3] Massaro, A., Calicchio, A., Maritati, V., Galiano, A., Birardi, V., Pellicani, L., Gutierrez Millan, M., Dalla Tezza, B., Bianchi, M., Vertua, G. & Puggioni, A. (2018) "A case study of innovation of an information communication system and upgrade of the knowledge base in industry by ESB, artificial intelligence, and big data system integration," *International Journal of Artificial Intelligence and Applications (IJAIA)*, Vol.9, No.5, pp 27-43.

[4] Massaro, A., Vitti, V., Lisco, P., Galiano, A., & Savino, N., (2019) "A business intelligence platform implemented in a big data system embedding data mining: a case of study," *International Journal of Data Mining & Knowledge Management Process (IJDKP)*, Vol.9, No.1, pp 1-20.

[5] Massaro, A., Leogrande, A., Lisco, P., Galiano, A., & Savino, N. (2019) "Innovative bi approaches and methodologies implementing a multilevel analytics platform based on data mining and analytical models: a case of study in roadside assistance services," *International Journal on Soft Computing, Artificial Intelligence and Applications* (IJSCAI), Vol.8, No.1, pp 17-36.

[6] Massaro, A., Mustich A., & Galiano, A. (2020) "Decision support system for multistore online sales based on priority rules and data mining," *Computer Science and Information Technology*, Vol. 8, No. 1, pp 1 - 12. doi: 10.13189/csit.2020.080101.

[7] Massaro, A., Meuli, G., & Galiano, A. (2018) "Intelligent electrical multi outlets controlled and activated by a data mining engine oriented to building electrical management," *International Journal on Soft Computing, Artificial Intelligence and Applications (IJSCAI)*, Vol. 7, No. 4, 2018, pp 1-20.

[8] Massaro, A., Meuli, G., Savino, N., & Galiano A. (2018) "A Precision agricolture DSS based on sensor treshold management for irrigation field," *Signal & Image Processing: An International Journal (SIPIJ)*, Vol. 9, No.6, pp 39-58.

[9] Massaro, A., Vitti, V., Galiano, A., & Morelli, A. (2019) "Business intelligence improved by data mining algorithms and big data systems: an overview of different tools applied in industrial research," *Computer Science and Information Technology*, Vol. 7, No.1, pp 1-21.

[9] Massaro, A., Galiano, A., Mustich, A., Convertini, D., Maritati, V., Colonna, A., Savino, N., Pace, A., & Iaquinta, L. (2019) "A case study of process engineering of operations in working sites through data mining and augmented reality," *International Journal of Data Mining & Knowledge Management Process (IJDKP)*, Vol.9, No.5, pp 1-20.

[10] Massaro, A., Galiano, A., Barbuzzi, D., Pellicani, L., Birardi, G., Romagno, D. D., & Frulli, L. (2017) "Joint activities of market basket analysis and product facing for business intelligence oriented on global distribution market: examples of data mining applications," (*IJCSIT) International Journal of Computer Science and Information Technologies*, Vol. 8, No. 2, pp 178-183.

[11] Massaro, A., Maritati, V., & Galiano, A. (2018) "Data mining model performance of sales predictive algorithms based on RapidMiner workflow," *International Journal of Computer Science & Information Technology (IJCSIT)*, Vol 10, No 3, pp 39-56.

[12] Massaro, A., Vitti, V., & Galiano, A. (2018) "Model of Multiple Artificial Neural Networks oriented on Sales Prediction and Product Shelf Design," *International Journal on Soft Computing, Artificial Intelligence and Applications (IJSCAI)*, Vol.7, No.3, pp1-19.

[13] Massaro, A., Maritati, V., Giannone, D., Convertini, D., & Galiano, A. (2019) "LSTM DSS automatism and dataset optimization for diabetes prediction," *Applied Sciences*, Vol. 9, No.17, 2019, 3532.

[14] Massaro, A., Maritati, V., Savino, N., Galiano, A., Convertini, D., De Fonte, E., & Di Muro, M., (2018) "A study of a health resources management platform integrating neural networks and DSS telemedicine for homecare assistance," *Information*, Vol. 9, No. 176, pp 1-20, doi:10.3390/info9070176.

[15] Krauth, E., Moonen, H., Popova, V., & Schut, M. (2005) "Performance measurement and control in logistics service providing," *ICEIS 2005 - Artificial Intelligence and Decision Support Systems*, pp. 239-247.

[16] Santos, S., Gouveia, J. B., & Gomes, P. (2004) "Measuring performance in supply chain - a framework," *Information Systems*, pp 1–6.

[17] Gamme, N. & Johansson, M. (2015) "Measuring supply chain performance through KPI identification and evaluation," *Master's thesis in "Supply Chain Management" and "Quality and Operations Management", Chalmers University of Technology*, Report No: E2015:109.

[18] Abdheen J., & Vimala, P. (2013) "Performance measurement and metrics in supply chain management," *Trends and Challenges in Global Business Management*, pp 139-150, ISBN 978-93-82338-84-0, 2013.

[19] Galiano, A., Massaro, A., Barbuzzi, D., Pellicani, L., Birardi, G., Boussahel, B., De Carlo, F., Calati, V., Lofano, G., Maffei, L., Solazzo, M., Custodero, V., Frulli, G., Frulli, E., Mancini, F., D'Alessandro, L., & Crudele, F. (2016) "Machine to Machine (M2M) open data system for business intelligence in products massive distribution oriented on Big Data," (*IJCSIT) International Journal of Computer Science and Information Technologies*, Vol. 7, No. 3, pp 1332-1336.

[20] Saponaro, A. (2018), "Cross-border risk assessment of earthquake-induced landslides in Central Asia," PhD Thesis 2018. Available online: https://depositonce.tu-berlin.de/bitstream/11303/7412/4/saponaro_annamaria.pdf

[21] Hochreiter, S., & Schmidhuber, J. (1997), "Long short-term memory," *Neural computation*, Vol. 9, No. 8, pp 1735–1780.

[22] Schafer, A., M. & Zimmermann, H.-G. (2007), "Recurrent Neural Networks are universal approximators," *International Journal of Neural Systems*, Vol. 17, No. 4, pp 253–263, DOI.

[23] Bianchi, F. M., Kampffmeyer, M., Maiorino, E. & Jenssen, R. (2017), "Temporal Overdrive Recurrent Neural Network," arXiv preprint arXiv:1701.05159.

[24] Dey, R., & Salemt, F. M. (2017) "Gate-variants of Gated Recurrent Unit (GRU) neural networks," *Proceeding of IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, Boston, MA, 2017, pp 1597-1600.

6. APPENDIX A: NEURAL NETWORK ALGORITHMS

The goal of the proposed paper is the study of motion detection sensitivity results of a prototype video surveillance system

```
# RNN Model construction
model = Sequential()
model.add(SimpleRNN(units=128, input_shape=(1, step), activation="relu",
return_sequences=True))
model.add(SimpleRNN(units=64, activation="relu", return_sequences=True))
model.add(SimpleRNN(units=64, activation="relu"))
model.add(Dense(32, activation="relu"))
model.add(Dense(32, activation="relu"))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer=Nadam(learning_rate=0.0001),
metrics=['cosine'])
model.summary()
# Model Training
start = timer()
rnn_h = model.fit(train_x, train_y, epochs=100, batch_size=128, verbose=0)
end = timer()
rnn_t = end - start
print('terminato.')
# Prediction (model behavior by considering separately the training and
# the testing dataset)
train_predict = model.predict(train_x)
test_predict = model.predict(test_x)
predicted = np.concatenate((train_predict, test_predict),axis=0)
# Output plot
# of initial data
index = df.index.values
plt.figure(figsize=(20,12))
plt.plot(index, df, label='Dati reali')
plt.plot(index + step, predicted, label='LSTM')
#plt.axvline(df.index[Tp], c="r")
plt.xlabel('Sequenza di tempi', fontsize=16)
plt.ylabel('Durata ciclo di conversione [giorni]', fontsize=16)
plt.xticks(fontsize=16)
```

```
plt.yticks(fontsize=16)
plt.legend(loc='upper right', shadow=True,frameon=False,fontsize=20)
plt.savefig('Durata_ciclo_conversione.png',bbox_inches='tight')
plt.show()
mse = np.sum(np.sqrt((predicted[:N-step]-df.values[step:])**2))/len(index[step:])
mse_smoothen = np.sum(np.sqrt((ndimage.gaussian_filter(predicted[:N-step], sigma=2.0,
order=0)-ndimage.gaussian_filter(df.values[step:], sigma=2.0, order=0))**2))/len(index[step:])
print(mse,mse_smoothen,len(index[step:]))

chisquared =np.sum(((predicted[:N-step]-df.values[step:])**2)/ predicted[:N-step])
print(chisquared)
# Model construction based on GRU cells
model_gru = Sequential()
model_gru.add(GRU(units=64, input_shape=(1, step), activation="relu",
return_sequences=True))
model_gru.add(GRU(units=64, activation="relu", return_sequences=True))
model_gru.add(GRU(units=64, activation="relu"))
model_gru.add(Dense(32, activation="relu"))
model_gru.add(Dense(32, activation="relu"))
model_gru.add(Dense(1))
model_gru.compile(loss='mean_squared_error', optimizer=Nadam(learning_rate=0.0001),
metrics=['cosine'])
model_gru.summary()
# Training of the GRU model
start = timer()
gru_h = model_gru.fit(train_x, train_y, epochs=100, batch_size=128, verbose=0)
end = timer()
gru_t = end - start
print('terminato.')
# Output analysis of the model
train_predict = model_gru.predict(train_x)
test_predict = model_gru.predict(test_x)
predicted_gru = np.concatenate((train_predict, test_predict),axis=0)
chisquared =np.sum(((predicted_gru[:N-step]-df.values[step:])**2)/ predicted_gru[:N-step])
from scipy.stats import chisquare
chisquared_ok = chisquare(f_obs=df.values[step:], f_exp=predicted_gru[:N-step],ddof=1899)
print(chisquared,chisquared_ok)
# Output of the model
# compared with initial results
index = df.index.values
plt.figure(figsize=(20,12))
plt.plot(index, df, label='Dati reali')
plt.plot(index + step, predicted_lstm,'r', label='LSTM')
plt.plot(index + step, predicted,'orange', label='RNN')
plt.plot(index + step, predicted_gru, 'm',label='GRU')
plt.xlabel('Sequenza di tempi', fontsize=16)
plt.ylabel('Durata ciclo di conversione [giorni]', fontsize=16)
```

```
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)
plt.legend(loc='upper right', shadow=True,frameon=False,fontsize=20)
plt.savefig('Durata_ciclo_conversione_rnn_lstm_gru.png',bbox_inches='tight')
plt.show()
index = df.index.values
plt.figure(figsize=(20,12))
plt.plot(index, ndimage.gaussian_filter(df, sigma=2.0, order=0), label='Dati reali')
plt.plot(index + step, ndimage.gaussian_filter(predicted_lstm, sigma=2.0, order=0),'r',
label='LSTM')
plt.plot(index + step, ndimage.gaussian_filter(predicted, sigma=2.0, order=0),'orange',
label='RNN')
plt.plot(index + step, ndimage.gaussian_filter(predicted_gru, sigma=2.0, order=0),
'm',label='GRU')
plt.xlabel('Sequenza di tempi', fontsize=16)
plt.ylabel('Durata ciclo di conversione [giorni]', fontsize=16)
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)
plt.legend(loc='upper right', shadow=True,frameon=False,fontsize=20)
plt.savefig('Durata_ciclo_conversione_rnn_lstm_gru_smooth.png',bbox_inches='tight')
plt.show()
# Detailed plot of the last 100 samples
# generated by the actual model
plt.figure(figsize=(20,12))
#plt.plot(index[900:], df[900:])
plt.plot(index[900:]+step, predicted[900:],'r', label='LSTM')
plt.plot(index[900:]+step, predicted_lstm[900:],'orange', label='RNN')
plt.plot(index[900:]+step, predicted_gru[900:],'m', label='GRU')
plt.xlabel('Sequenza di tempi', fontsize=16)
plt.ylabel('Durata ciclo di conversione [giorni]', fontsize=16)
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)
plt.legend(loc='upper right', shadow=True,frameon=False,fontsize=20)
plt.savefig('Durata_ciclo_conversione_rnn_lstm_gru_zoom.png',bbox_inches='tight')
plt.show()
# Loss plot of the 3 implemented models
#f, ax = plt.subplots()
#f = plt.figure()
plt.figure(figsize=(20,12))
#ax.set_title('MSE / Epoche')
plt.plot(rnn_h.history['loss'], label='RNN')
plt.plot(lstm_h.history['loss'], label='LSTM')
plt.plot(gru_h.history['loss'], label='GRU')
plt.yscale('log')
plt.legend(loc='upper right', shadow=True,frameon=False,fontsize=20)
#ax.legend(['RNN', 'LSTM', 'GRU'], loc = 0)
plt.ylabel('Funzione di Loss', fontsize=16)
plt.xticks(fontsize=16)
```

```
plt.yticks(fontsize=16)
plt.legend(loc='upper right', shadow=True,frameon=False,fontsize=20)
plt.savefig('loss.png',bbox_inches='tight')
plt.show()
# Accuracy plot
#f, ax = plt.subplots()
plt.figure(figsize=(20,12))
plt.plot(rnn_h.history['cosine'],'r', label='LSTM')
plt.plot(lstm_h.history['cosine'],'orange', label='RNN')
plt.plot(gru_h.history['cosine'],'m', label='GRU')
#ax.legend(['RNN', 'LSTM', 'GRU'], loc = 0)
plt.ylabel('Funzione di Loss', fontsize=16)
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)
plt.legend(loc='lower right', shadow=True,frameon=False,fontsize=20)
plt.savefig('cosine.png',bbox_inches='tight')
plt.show()
```

7. APPENDIX B: CASSANDRA BIG DATA NODE CONFIGURATION AND TESTING DATASET STORAGE

The testing dataset are collected into a Cassandra NoSQL Big Data system. The script used for the table creation is as follows:

```
create table Key_performance_indicators (id timeuuid PRIMARY KEY, ontime_analysis float,
lead_time_di_approvviggionamento float, durata_ciclo_conversione float,
percentuale_nuovi_clienti float, number_of_late_deliveries int, velocita_chiusura_trattativa float,
attivita_completate_trattativa int, ratio_attivita_settore_completate_totale_attivita_completate
float, ratio_attivita_settore_non_completate_totale_attivita_non_completate float,
valutazione_rischio float, referenze_acquisite int, variabilita_servizi_venduti int,
order_processing_time float, percentuale_incassato float, num_contratti_chiusi
int,tasso_conversione int, ricorrenza_vendita int);
```

Below is illustrated a screenshot proving the correct implementation of the tables of the prototype platform.

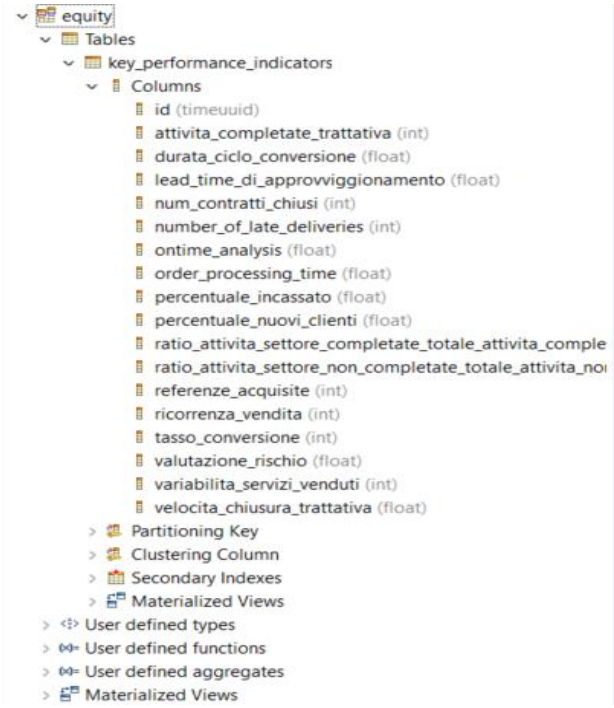

Figure 13. Screenshot proving the correct implementation of the table into the prototype platform.

The dataset model has been written into Cassandra by executing the following script:

```
import pandas as pd
from cassandra.cluster import Cluster
from uuid import uuid1
import math

N = 1000  #
Tp = 900

t = np.arange(0, N)

data = []
for i in range(N):

   ontime_analysis = 0.1 * random.randrange(0, 2) * np.sin(0.001 * random.randrange(5, 10) * t[i]) + 0.2 * np.sin(0.03 * t[i])
   lead_time_di_approvviggionamento = max(0.1 * random.randrange(0, 4) * np.sin(0.005 * random.randrange(5, 10) * t[i]),
            0.1 * random.randrange(0, 2) - value)
   durata_ciclo_conversione = 0.1 * random.randrange(0, 5) * np.cos(0.001 * random.randrange(5, 10) * t[i]) + 0.1 * random.randrange(0, 5) * np.cos(0.001 * random.randrange(3, 7) * t[i])
```

```
    percentuale_nuovi_clienti = 0.1 * random.randrange(0, 3) * np.cos(0.005 *
random.randrange(2, 8) * t[i]) + 0.8 * random.randrange(0, 5) * np.cos(
        0.001 * random.randrange(3, 9) * t[i])
    number_of_late_deliveries = int(0.1 * random.randrange(0, 8) * np.cos(0.003 *
random.randrange(1, 8) * t[i]) + 0.5 * random.randrange(0, 1) * np.cos(
        0.001 * random.randrange(3, 8) * t[i]))
    velocita_chiusura_trattativa = 0.1 * random.randrange(0, 1) * np.sin(0.004 *
random.randrange(5, 9) * t[i]) + 0.1 * random.randrange(0, 2) * np.cos(
        0.001 * random.randrange(3, 6) * t[i])
    attivita_completate_trattativa = int(0.1 * random.randrange(0, 9) * np.cos(0.007 *
random.randrange(1, 7) * t[i]) + 0.8 * random.randrange(0, 3) * np.cos(
        0.001 * random.randrange(3, 5) * t[i]))
    ratio_attivita_settore_completate_totale_attivita_completate = 0.1 * random.randrange(0, 3)
* np.tan(0.009 * random.randrange(2, 7) * t[i]) + 0.7 * random.randrange(0, 4) * np.tan(
        0.001 * random.randrange(3, 4) * t[i])
    ratio_attivita_settore_non_completate_totale_attivita_non_completate = 0.1 *
random.randrange(0, 4) * np.cos(0.006 * random.randrange(1, 8) * t[i]) + 0.2 *
random.randrange(0, 8) * np.tan(
        0.001 * random.randrange(3, 7) * t[i])
    valutazione_rischio = 0.1 * random.randrange(0, 2) * np.sin(0.008 * random.randrange(2, 9)
* t[i]) + 0.4 * random.randrange(0, 9) * np.cos(
        0.001 * random.randrange(3, 4) * t[i])
    referenze_acquisite = int(0.1 * random.randrange(0, 3) * np.cos(0.002 * random.randrange(5,
8) * t[i]) + 0.6 * random.randrange(0, 4) * np.sin(
        0.001 * random.randrange(3, 9) * t[i]))
    variabilita_servizi_venduti = int(0.1 * random.randrange(0, 7) * np.sin(0.001 *
random.randrange(1, 2) * t[i]) + 0.5 * random.randrange(0, 6) * np.sin(
        0.001 * random.randrange(3, 8) * t[i]))
    order_processing_time = 0.1 * random.randrange(0, 6) * np.sin(0.003 * random.randrange(2,
3) * t[i]) + 0.4 * random.randrange(0, 3) * np.cos(
        0.001 * random.randrange(3, 4) * t[i])
    percentuale_incassato = 0.1 * random.randrange(0, 5) * np.cos(0.008 * random.randrange(2,
9) * t[i]) + 0.3 * random.randrange(0, 2) * np.sin(
        0.001 * random.randrange(3, 9) * t[i])
    num_contratti_chiusi = int(0.1 * random.randrange(0, 4) * np.tan(0.007 *
random.randrange(1, 6) * t[i]) + 0.9 * random.randrange(0, 1) * np.cos(
        0.001 * random.randrange(3, 8) * t[i]))
    tasso_conversione = 0.1 * random.randrange(0, 8) * np.cos(0.003 * random.randrange(3, 8) *
t[i]) + 0.2 * random.randrange(0, 5) * np.sin(
        0.001 * random.randrange(3, 9) * t[i])
    ricorrenza_vendita = int(0.1 * random.randrange(0, 9) * np.sin(0.001 * random.randrange(2,
7) * t[i]) + 0.1 * random.randrange(0, 7) * np.cos(
        0.001 * random.randrange(3, 5) * t[i]))
    data.append([ontime_analysis, lead_time_di_approvviggionamento,
durata_ciclo_conversione, percentuale_nuovi_clienti, number_of_late_deliveries,
velocita_chiusura_trattativa,
```

attivita_completate_trattativa,
ratio_attivita_settore_completate_totale_attivita_completate,
ratio_attivita_settore_non_completate_totale_attivita_non_completate, valutazione_rischio,
referenze_acquisite, variabilita_servizi_venduti, order_processing_time,
percentuale_incassato,num_contratti_chiusi, tasso_conversione, ricorrenza_vendita])
data = np.array(data)

Author

Alessandro Massaro (corresponding author): Professor Alessandro Massaro (ING/INF/01, FIS/01, FIS/03) carried out scientific research at the Polytechnic University of Marche, at CNR, and at Italian Institute of Technology (IIT) as Team Leader by activating laboratories for nanocomposite sensors for industrial robotics. He is in MIUR register as scientific expert in competitive Industrial Research and social development, and he is currently head of the Research and Development section and scientific director of MIUR Research Institute Dyrecta Lab Srl. Member of the International Scientific Committee of Measurers IMEKO and IEEE Senior member, recently received an award from the National Council of Engineers as Best Engineer of Italy 2018 (Top Young Engineer 2018).

Giovanni Dipierro. Giovanni Dipierro obtained a Master's Degrees in Physics from the Univesity of Milan (Italy) in 2014. In 2015 he was appointed as a PhD student in physics, astrophysics and applied physics at University of Milan, earning his PhD in 2017. After receiving his doctored, he worked as post-doc researcher at University of Leicester (UK). His research focus on developing artificial intelligence algorithms to analyse data and performing numerical simulations of fluid dynamics and structural mechanics over a large range of physical phenomena. In 2019 he joined the research team of Dyrecta Lab Srl in Conversano (Ba), Italy.

Annamaria Saponaro. Annamaria Saponaro achieved her Bachelor's and Master's Degrees in Geoscience in Italy. In 2012 she was appointed as a PhD researcher at the German Research Center for Geosciences in Potsdam, Germany, and in 2017, she received her PhD from the Technical University in Berlin, Germany. Her research work focused on risk assessment in the context of natural hazards. Besides carrying out research activities she covered the role of risk analyst for insurance companies. In 2019 she joined the research team of Dyrecta Lab Srl in Conversano (Ba), Italy.

Angelo Maurizio Galiano. Angelo Maurizio Galiano is CEO at Dyrecta Lab Srl - research institute accredited by the Italian Ministry of University and Scientific Research. He has more than 20 years of experience in the field of Information Technologies. He received M.S. degree in Education Science in 2009. His current research interests include neural networks, smart health and predictive analytics.