ᲡᲐᲥᲐᲠᲗᲕᲔᲚᲝᲡ ᲛᲔᲪᲜᲘᲔᲠᲔᲑᲐᲗᲐ ᲔᲠᲝᲕᲜᲣᲚᲘ ᲐᲙᲐᲓᲔᲛᲘᲘᲡ ᲛᲝᲐᲛᲑᲔ, ტ. 12, №2, 2018
BULLETIN OF THE GEORGIAN NATIONAL ACADEMY OF SCIENCES, vol. 12, no.2, 2018

Informatics

Emergency Brigade Selection for Improvement of Emergency Medical Service Dispatching

Giorgi Kiviladze

Georgian Technical University (GTU); Emergency and Operative Response Center "112", Tbilisi, Georgia

(Presented by Academy Member Giorgi Gogichaishvili)

ABSTRACT. **The work represents the project of cooperation with the Emergency and Operative Response Center - MIA LEPL "112", the ways of development of data analysis for improvement of emergency brigade dispatching. To form and analyze the dataset, programming language Python and collection of machine learning algorithms Weka toolbox are applied. The architecture of Artificial Neural Network is created. The project aims to help "112" dispatchers to make an optimal decision and decrease the response time. The Emergency and Operative Response Center "112" is ready to use the results of the project for improvement of emergency service and supports the research with the non-confidential information, necessary to fulfill the project. "112" is ready to take into consideration the recommendations based on the research.** *© 2018 Bull. Georg. Natl. Acad. Sci.*

Key words: Emergency Medical Service, dispatching, dataset, algorithm

The research is aimed to develop the data analysis and machine learning system to provide optimization of emergency brigade dispatching.

Emergency case messages usually enter the center "112". According to the message content a call-taker operator performs the following: gets and processes the message information mentally to reveal its main part (Incident Type) that automatically checks the type of help needed in the case (Case Category).

It ought to be mentioned that more than one service categories could be checked for each case. For instance, Medical and Police are checked automatically after assigning incident type as "Wound".

Currently call-takers use the following model. Different medical incident types are classified by professional medics in three fixed groups of "low", "medium" and "high" urgency.

Afterwards the registered information is sent to all checked service dispatchers.

As our recent research is aimed only at the optimization of medical emergency service, the work of medical dispatchers is discussed below.

Dispatchers of each region see the list of recently entered active cases along with the set of ambulance cars available for the moment. Each case remains currently active until a dispatcher attaches it for the following response. As to the "available" ambulance cars, several statuses are defined, such as "free", "with attached case", "gone to the address", "arrived at the address", "in hospitalization process" and a few other statuses connected with technical services.

Besides, it should be mentioned that all the cases are classified according to their priorities of urgency ("low", "medium" or "high") due to their incident type (the list of incident types and their priorities is fixed).

The task of dispatcher is to reveal the most urgent case from the active case list and connect it to the ambulance car chosen as the most proper for the fastest responding. Such task is often rather difficult to perform, having in mind human factors of being tired, stressed, feeling rushed and so on [1,2]. It is decided to organize the automated help for emergency dispatchers to achieve optimal decision of dispatching problems.

Brigade Selection Problem

The aim of the current research is to provide the Emergency and Operative Response Center "112" with the methods for solving the brigade selection problem - help the dispatcher of "112" to make a proper decision and decrease the response time.

The problem is caused by lack of information about the approximate time needed by the brigade for fulfilling the current task and arriving at the new address. For example, a "free" emergency brigade is in a forty minutes ride from the call initiator, but another brigade having got the task nearby can be free in about 5 minutes. It is clear that suboptimal management can cause time delays, more financial expenses and more human resource. Choosing the appropriate brigade is one of the most important parts of the emergency case flow.

At first, let us consider the optimization of the brigade selection process.

It begins with calculating the probability of patient hospitalization. If the probability of hospitalization is high ($P \geq 0.5$) the occupied ambulance car is not considered for future suggestion at that moment. In the case of low probability of hospitalization ($P < 0.5$) the algorithm estimates the probable time of occupation of the ambulance car and provides the dispatcher with this information. So this ambulance car could be considered to be free after that time. It should be mentioned that the algorithm could give only recommendations which are to be considered and checked by the dispatcher contacting directly with the suggested ambulance medics.

The probability of patient hospitalization as well as the approximate time to be spent by the brigade with the patient is calculated considering the following facts: incident type, age and gender of a patient, priority of the case, whether the emergency services other than medical (police, fire and rescue) are needed. As a base of the research, the information about the set of former cases and the facts of hospitalisation of the patients is used. The data collected since foundation of MIA LEPL "112" are used. A large dataset of emergency cases including all the important parameters like the patient data and timestamps of brigade status changes is received.

Besides, it is important to estimate the ride time of an ambulance. The already existing routing services could be used for that. One option is to use Google traffic to indicate the approximate time to get from one point to another, considering traffic jams [3].

Dataset preparation work is presented in two steps: data-gathering and data pre-processing.

First of all, the data of the last five years are selected in CSV format.

Then the data are processed using Python programming: remove anomalies and prepare the data for further steps.

Data-gathering

The current project is closely linked with the Emergency and Operative Response Center "112". This means that the actual data as well as the database structure is confidential.

The description of the first step of dataset preparation is listed below:

- CASE_ID - the unique code of a case (needed only on the data-gathering step).
- CALL_ID - the unique code of a call (needed only on the data-gathering step).
- NUMBER – call initiator phone number (needed only on the data-gathering step).
- DATE_CREATED – call entering timestamp.
- CASE_TEMPLATE_ID – identificator of the main incident type (heart problem, convulsion, labored breathing...).
- PATIENT_AGE_YEARS,
  PATIENT_AGE_MONTHS,
  PATIENT_AGE_DAYS,
  the age of a patient in years, months or days (Fig.1).

Fig. 1. The age of a patient.

  A call taker usually fills the case-card presenting the age of a patient in days if the patient is a newborn. The patient's age is fixed in months in case of infants and the age is shown in years for all other patients.

- PATIENT _GENDER - patient's gender.
- CATEGORIES – the parameter shows information about all urgent services included in the incident management (emergency, fire/rescue, police…).
- SPENT_MINS – the time spent by ambulance brigade on the patient (in case of hospitalisation the riding time is added to the time spent at the patient's address).
- PREV_HOSP_COUNT – count of hospitalisation cases with the same incident type called from the same number within a year.
- LASTHOSPDD – the time (the number of days) passed after the last hospitalisation of the patient for the last year (the incident type as well as the call number are the same).
- CALL_COUNT – the number of cases registered from the same phone number within the last month.

Data Preprocessing

Data pre-processing is an important step in preparing the data for analysis. Data gathering methods are often loosely controlled, resulting in out-of-range values (e.g., SPENT_MINS is a negative number), unusable data (e.g. calls made without sim-cards generate thousands of unusable records), missing values, etc. Analyzing data that is not curated carefully can produce misleading results. Thus, proper representation and quality of data is first and foremost before running analysis [4]. Often, data pre-processing is the most important phase of a machine learning project [5].

Python programming is used for dataset preparation and analysis.`

At first, triage cases (cases where several ambulances were needed simultaneously) and unfilled data were removed:

```
# The parameter keep=False provides complete removal of duplicated records (multiple cases on the same call).
df_orig = df_orig.drop_duplicates(subset=['CALL_ID'], keep=False)

#It is important that there are no blanks in the following columns.
# Using dropna method pointed with how='any' parameter, all the rows where above mentioned columns with blanks are placed are removed completely.
df_orig.dropna(subset=['CASE_ID', 'CALL_ID', 'DATE_CREATED',
'CASE_TEMPLATE_ID', 'NOM', 'CATEGORIES', 'SPENT_MINS'],
how='any', inplace=True
```

The further description of the dataset pre-processing is listed below:

- DATE_CREATED – call entering timestamp underwent categorisation according to time ranges special for MIA LEPL “112”:
  TIME_02_08, TIME_08_12, TIME_12_17, TIME_17_23, TIME_23_02. Particularly, the date was removed from the timestamp and categorization using One Hot Encoding method was held on the remaining time parameters. That means only 1 column from 5 filled with cipher “1” for each line while others are filled with “0”.
  On the current stage of research, only these time ranges are used while in the future paying attention to the seasons is planned as well.
- CASE_TEMPLATE_ID – the data underwent categorisation procedure using One Hot Encoding method. See the code pattern below:

```
pd.get_dummies(df, columns=["CASE_TEMPLATE_ID"], prefix='CTID')
```

  As a result, 194 unique columns were got, only one of them being filled with “1” for each case, while others were “0”.
- PATIENT_AGE – Two methods of processing age data are used:
  - First of all, anomalies were removed. Then age categorization was carried out using One Hot Encoding method, the following age ranges chosen as categories: [0 – 40] days, [41 day – 1 year], (1- 6] years, (6 – 15] years, (15 – 60) years, [> 60] years.
  - We try to move in another direction as well: instead of age categorisation all age data are fixed as numbers of days and the data are normalized between the range 0 and 1 using the following method:

$$X_n = (X – X_{min}) / (X_{max} – X_{min}), \tag{1}$$

  X_n meaning the normalized value of the patient’s age.
- PATIENT _GENDER – if the gender of a patient was not provided in the case-card, the mean data equal to 0.43 were taken instead of NaN values. See the code below:

```
df['PATIENT_GENDER'].fillna(df['PATIENT_GENDER'].mean(), inplace= True
```

- CATEGORIES - The CATEGORIES column was split in separate records from the coma separated CATEGORIE_ID values and the original column was turned into 8 unique columns according to 8 different services (ambulance, fire, police, criminal and so on). All the columns were filled with "1" where the relevant category was concerned. The others where filled with “0”.
  The categorisation procedure was held using the following code:

```
df_CAT_TMP=df_orig.set_index('CASE_ID').CATEGORIES.str.split(r',',
  expand=True).stack().reset_index(level=1,
```

```
drop=True).to_frame('CATEGORIES')
df_only_CAT=pd.get_dummies(df_CAT_TMP, prefix='CAT',
  columns=['CATEGORIES']).groupby(level=0).sum()
df = pd.merge(df_with_CTID, df_only_CAT, on=df_with_CTID.CASE_ID,
  how='inner')#join two df
```

- SPENT_MINS – unchanged.
- PREV_HOSP_COUNT – unchanged.
- LASTHOSPDD – unchanged.
- CALL_COUNT – unchanged.
- HOSPITALISATION - if the patient was hospitalized, “1” was put in HOSPITALISATION column. “0” goes if the answer is negative.

  In case of hospitalisation the riding time is added to the time spent at the patient’s address and so the value of SPENT_MINS time is got.

  See the code below that generates the HOSPITALISATION column:

```
#If the date of the last hospitalization is the same as the date of current call we assume that the case was ended with hospitalization.
df['HOSPITALISATION']=np.where(df['LASTHOSPDD']==0,'1','0')
# In order to avoid double counting, in above mentioned cases we diminish the PREV_HOSP_COUNT by 1.
for i, row i df.iterrows():
   if row['LASTHOSPDD']==0:
df.loc[i,'PREV_HOSP_COUNT'] -=1
df.loc[i,'LASTHOSPDD'] = np.nan
if df.loc[i,'PREV_HOSP_COUNT']==0:
df.loc[i,'PREV_HOSP_COUNT'] = np.nan
```

The result of data pre-processing is the set of 219 column and about 2.7 milion rows data ready for further studies. It is to be mentioned that in order to use various algorithms the dataset may need certain modifications.

Data Visualization and Analysis

Further work on the dataset is carried out simultaneously in two directions: Python programming and Weka toolbox.

Weka is a collection of machine learning algorithms for data mining tasks. It contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization.

To demonstrate the correlation of various data Weka visualization tools were used. As an example of visualization of the data, the relationship between the time spent by emergency brigade with the patient and the count of former hospitalizations of the patient with the same incident type and the same phone number is presented below (Fig. 2). Each point in Fig. 2 represents a random case. The orange color indicates that the case ended with hospitalization. The blue color of the point shows that the patient was not hospitalized

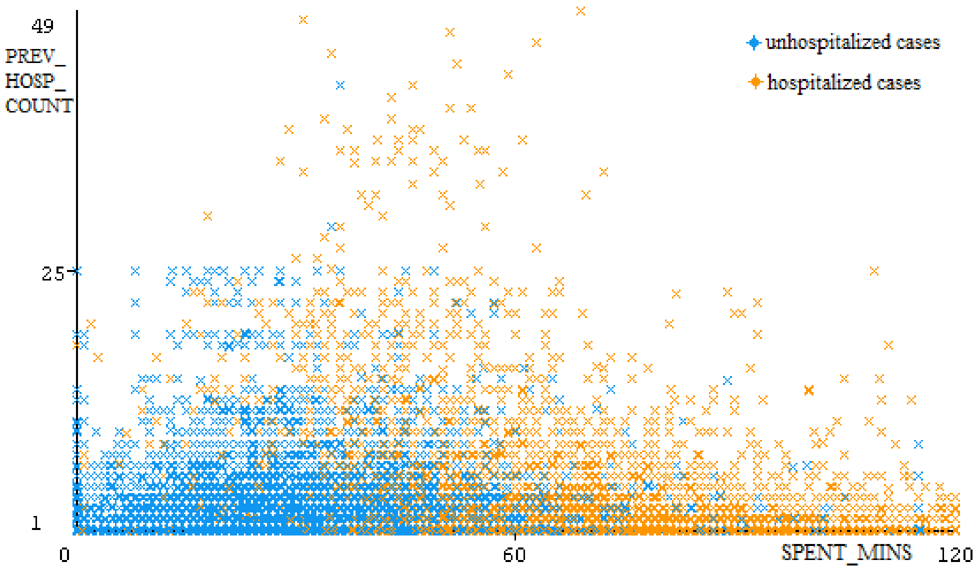

Fig. 2. Visualization of relationship between the data: SPENT_MINS, PREV_HOSP_COUNT and HOSPITALIZATION.

As it is seen from the visualization stage, the data are largely dispersed. It was decided to use the last 5-year data to form the dataset.

Estimation of the Time Spent by the Ambulance Brigade

The project of estimating the time spent by the ambulance brigade on each case is divided into two sub problems.

Classification problem:

Answer: HOSPITALIZATION.

Calculate the probability of hospitalization of a patient in each case. Having in mind that the decision about patient's hospitalization is always made by an ambulance doctor, it is impossible for "112" dispatcher to know for sure beforehand whether the hospitalization of a patient will take place. So, the output of the algorithm is the value of hospitalization probability. The approach of Artificial Neural Network is to be applied to work at the problem (Fig. 3).

The input parameters are worked out on the data preprocessing stage (see "Data preprocessing" above in the text). 194 different "CASE_TEMPLATE_ID" input parameters are shown in the scheme as one parameter "Incid. type", parameter "Age" means 6 input ranges and so on, as it is described above.

As we have no information about the time needed for current patient case, all the columns except SPENT_MINS one are present in the input layer.

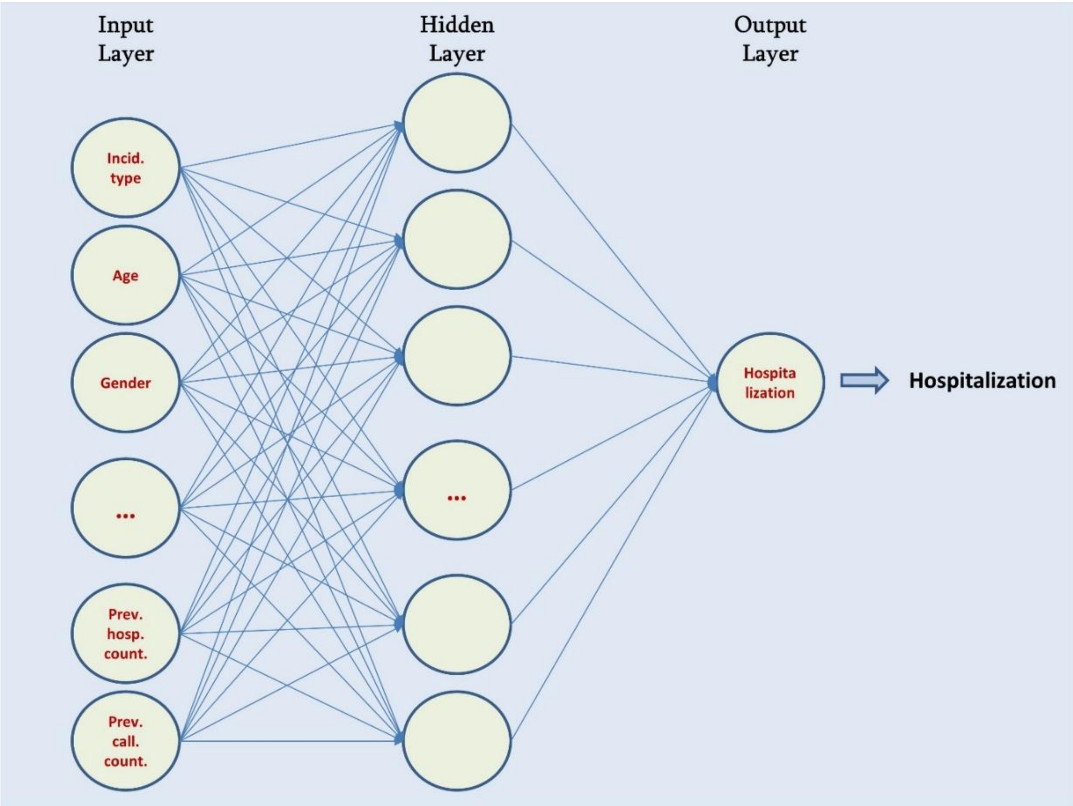

Fig. 3. Calculation of probability of patient's hospitalization based on the parameters worked out on the data preprocessing stage.

Regression Problem:

Answer: SPENT_MINS.

Estimate the time spent by the ambulance brigade on a current case patient if the probability of hospitalization P<0.5.

The training set consists of all the columns while the rows corresponding to the probably hospitalized (P ≥ 0.5) cases are not included (Fig. 4).

Only three ranges of time probably spent by the ambulance brigade at the patient's address are shown in the output layer. Working with more time ranges would increase the probability of errors significantly.

To reveal the output layer results, most proximal to the real time values spent by ambulance brigades on the cases, it is necessary to work out optimal architecture of the hidden layer [6].

At the begining of research it is hard to forsee all the technologies and tasks optimal for work. The existing point of view on the above mentioned issues and methodology can be readily changed in the work process.

Considering the scale of the work, the cooperation with the "112" center is to be continued during the following years.

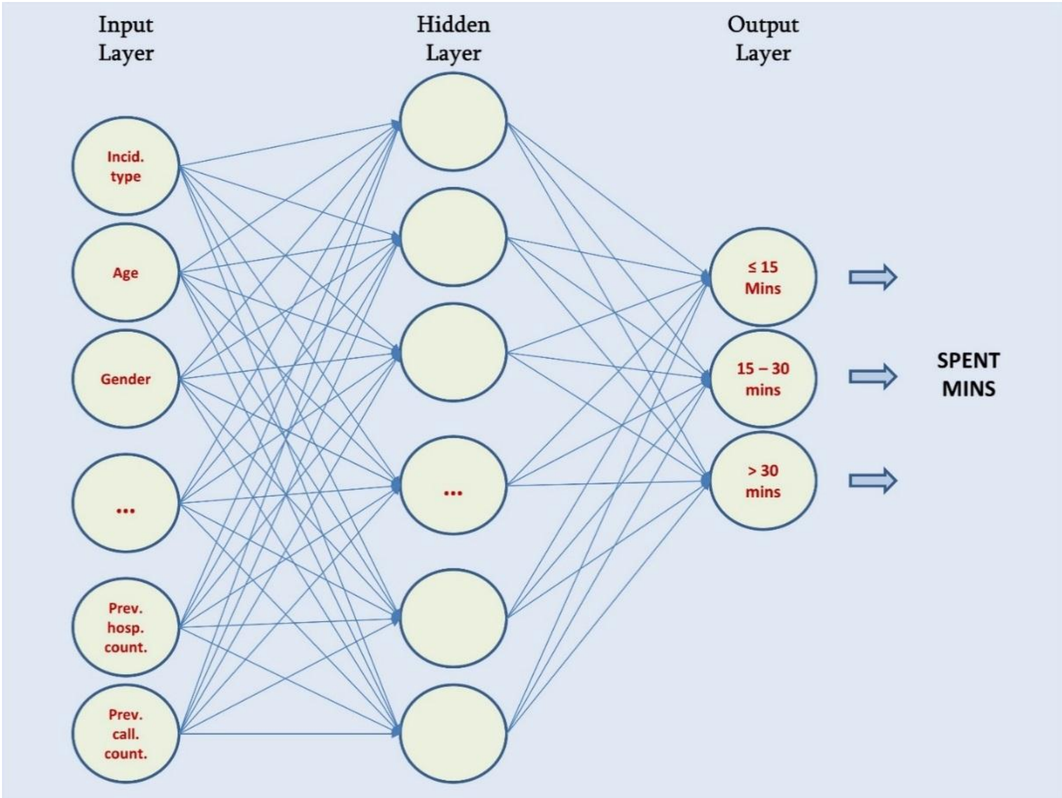

Fig. 4. The scheme of estimation of time spent by ambulance brigade at the patient's address, if the probability of hospitalization is low ($P < 0.5$).

Conclusion

At the first stage of the research the following work was done. The medical emergency service challenges were analyzed. The problems for applying computing methods were revealed. The dataset was formed on the base of last 5-year medical emergency cases. Special system was created to help resolving increased problems of emergency medical service, aiming to make easier the emergency dispatchers' work.

Acknowledgment

This research was supported by Shota Rustaveli National Science Foundation (SRNSF) [PhDF2016_219]. I am thankful to my supervisor Prof. Gia Surguladze and obliged to my colleagues at Georgian Emergency and Operative Response Center - MIA LEPL "112" for their kind support. I am thankful to my consultants in Finland (University of Tampere) Prof. Jyrki Nummenmaa and Germany (Friedrich-Alexander University Erlangen-Nürnberg) Prof. Klaus Meyer-Wegener for many professional advices and help in my work. I am especially thankful to my Finnish colleague and friend Aapo Koski for interesting scientific discussions and advices he kindly gave me that helped much in my further research.

ინფორმატიკა

გადაუდებელი დახმარების ბრიგადების შერჩევის ამოცანა სასწრაფო სამედიცინო მომსახურების დისპეტჩერიზაციის გაუმჯობესების მიზნით

გ. კივილაძე

საქართველოს ტექნიკური უნივერსიტეტი; გადაუდებელი დახმარების ოპერატიული მართვის ცენტრი "112", თბილისი, საქართველო

(წარმოდგენილია აკადემიის წევრის გ. გოგიჩაიშვილის მიერ)

აღწერილია გადაუდებელი დახმარების ოპერატიული მართვის ცენტრ შსს სსიპ „112"-თან თანამშრომლობის პროექტი. შემუშავებულია მონაცემთა ანალიზის სისტემა სასწრაფო დახმარების ბრიგადების შერჩევის პროცესის გაუმჯობესებისათვის. მონაცემთა სიმრავლის ფორმირებისა და ანალიზისათვის გამოყენებულია მულტიპარადიგმული ენა Python და მანქანური დასწავლის ალგორითმების ნაკრები Weka, შემუშავებულია ხელოვნური ნეირონული ქსელის არქიტექტურა. პროექტის მიზანია, დაეხმაროს „112"-ის დისპეტჩერებს ოპტიმალური გადაწყვეტილების სწრაფად მიღებაში. „112" მხარს უჭერს ამ კვლევებს, უზრუნველყოფს რა, არაკონფიდენციალური ინფორმაციით და მზად არის, გაითვალისწინოს კვლევის შედეგებზე დაფუძნებული რეკომენდაციები გადაუდებელი სამედიცინო სამსახურის მუშაობის გაუმჯობესების მიზნით.

REFERENCES

1. Bedini S., Braun F., Weibel L. et al. (2017) Stress and salivary cortisol in emergency medical dispatchers: A randomized shifts control trial. PLoS ONE. 12(5): e0177094. doi:10.1371/journal.pone.0177094.
2. Mew E.J., Ritchie S.D., VanderBurgh D. et al. (2017) An environmental scan of emergency response systems and services in remote First Nations communities in Northern Ontario. International Journal of Circumpolar Health.76(1): 1320208. doi:10.1080/22423982.2017.1320208.
3. Lee C.L., Huang C.Y., Hsiao T.C. et al. (2014) Impact of vehicular networks on emergency medical services in urban areas. International Journal of Environmental Research and Public Health.11(11): 11348-11370. doi:10.3390/ijerph111111348.
4. Luo G. (2015) MLBCD: a machine learning tool for big clinical data. Health Information Science and Systems, 3(3). doi:10.1186/s13755-015-0011-0.
5. Chicco D. (2017) Ten quick tips for machine learning in computational biology. *BioData Mining*, **10** (35): 1–17. doi:10.1186/s13040-017-0155-3.
6. Manning T., Sleator R.D., Walsh P. (2014) Biologically inspired intelligent decision making: A commentary on the use of artificial neural networks in bioinformatics. *Bioengineered*, 5(2): 80-95. doi:10.4161/bioe.26997.

Received March, 2018