Long Short-Term Memory for Predicting Firemen Interventions

Selene Leya Cerna Ñahuis[1], Christophe Guyeux[2], Héber Hwang Arcolezi[1], Raphaël Couturier[2],
Guillaume Royer[3] and Anna Diva Plasencia Lotufo[1]

Abstract—**Many environmental, economic and societal factors are leading fire brigades to be increasingly solicited, and they, therefore, face an ever-increasing number of interventions, most of the time with constant resources. On the other hand, these interventions are directly related to human activity, which itself is predictable: swimming pool drownings occur in summer while road accidents due to ice storms occur in winter. One solution to improve the response of firefighters with constant resources is therefore to predict their workload, i.e., their number of interventions per hour, based on explanatory variables conditioning human activity. The purpose of this article is to show that these interventions can indeed be predicted, in a nonabsurd way, from state-of-the-art tools such as recurrent long short-term memory neural networks (LSTM). From the list of interventions in the Doubs (France), we show that it is possible to build, from scratch, a neural network capable of reasonably predicting the interventions of 2017 from those of 2012-2016. While the results could be improved, they are already promising and would allow the actions of firefighters with a constant resource to be optimized.**

I. Introduction

Fire departments in the world constantly seek strategies to decrease their response time to interventions since it is one of the most important factors to consider; their ability to save more lives and rescue people depends on it. The data collected through the years about their interventions during fires, road accidents, and other types of incidents could be used to develop a data-driven decision making approach in order to understand the trends of certain events such as knowing the hours following a particular event the number of likely interventions ensuing as well as their type or place. In this way, the efficiency of their response could increase and their operational costs would be decreased.

Moreover, nature has inspired most of the inventions that humans have developed and brain's architecture is an example of inspiration on how to build an intelligent machine. We learn considering our memories, we comprehend contexts based on our understanding of previous situations and we can complete phrases using first words as a basis. These characteristics are well represented in recurrent

neural networks with Long Short-Term Memory (LSTM), which have emerged as effective and scalable model for several learning problems related to sequential data (e.g., handwriting recognition, speech recognition, human activity recognition and traffic prediction), and it does not suffer from the effect of the vanishing or exploding gradient problem as simple recurrent networks do [1]. One of the central keys behind LSTM success is its memory cell, which can maintain its state over time by learning what to store in the long-term state, what to throw away and what to read from it by passing on nonlinear gating units regulating the information flow into and out of the cell. Fig. 1 shows in more details how the long-term state $c_{(t-1)}$ initially goes through a forget gate, dropping out some memories, and then it adds some new memories that were properly selected by an input gate. The result $c_{(t)}$ is sent straight out without any further transformation, i.e., at each time step, some memories are dropped and others are added. Furthermore, after passing by the input gate, the long-term state is copied and passed through an activation function $tanh$, and finally the result is filtered by an output gate. This procedure generates the short-term state $h(t)$ that is equal to the cell's output for this time step $y(t)$.

Mathematically, it is expressed as:

$$i_{(t)} = \sigma(W_{xi}^T \cdot x_{(t)} + W_{hi}^T \cdot h_{(t-1)} + b_i) \qquad (1)$$

$$f_{(t)} = \sigma(W_{xf}^T \cdot x_{(t)} + W_{hf}^T \cdot h_{(t-1)} + b_f) \qquad (2)$$

$$o_{(t)} = \sigma(W_{xo}^T \cdot x_{(t)} + W_{ho}^T \cdot h_{(t-1)} + b_o) \qquad (3)$$

$$g_{(t)} = \tanh(W_{xg}^T \cdot x_{(t)} + W_{hg}^T \cdot h_{(t-1)} + b_g) \qquad (4)$$

$$c_{(t)} = f_{(t)} \otimes c_{(t-1)} + i_{(t)} \otimes g_{(t)} \qquad (5)$$

$$y_{(t)} = h_{(t)} = o_{(t)} \otimes \tanh(c_{(t)}) \qquad (6)$$

where W_{xi}, W_{xf}, W_{xo} and W_{xg} are the weight matrices of each four layers for their connection to the input vector $x_{(t)}$; W_{hi}, W_{hf}, W_{ho} and W_{hg} are the weight matrices of each four layers for their connection to the previous short-term state $h_{(t-1)}$; and b_f, b_g, b_i and b_o are the bias terms of each of the four layers. For more details about LSTM architecture, interested readers are strongly advised to read and consult [1]–[4].

To the best of the authors' knowledge, researches based on the forecast of the number of interventions, response speed, materials and engine used by fire departments are scarce in the literature. In particular, the use of LSTM neural networks to predict the number of firefighters interventions is provided for the first time through this paper. The data collected from

[1]S. L. C. Ñahuis, H. H. Arcolezi and A. D. P. Lotufo are with Department of Electrical Engineering, São Paulo State University, UNESP, Ilha Solteira, São Paulo, Brazil. e-mail: selene.cerna@gmail.com, heber-hwang@gmail.com, anna.lotufo@unesp.br

[2]C. Guyeux and R. Couturier are with FEMTO-ST Institute, Univ. Bourgogne Franche-Comte (UBFC), CNRS, France. email: christophe.guyeux@univ-fcomte.fr, raphael.couturier@univ-fcomte.fr

[3]G. Royer is with SDIS 25 - Service Départemental d'Incendie et de Secours du Doubs, France. email: Guillaume.ROYER@sdis25.fr
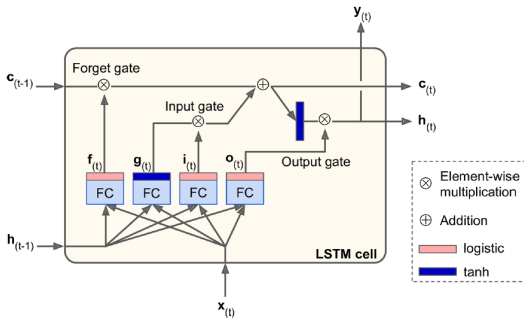
Fig. 1: LSTM Cell [3]

firefighters interventions is considered as a sequential data set that will feed our LSTM model to afterward predict the number of future incidents. To complete our data set, several factors are included such as weather, traffic, holidays, the height of the rivers, disease statistics, etc. Therefore, this will help firefighters to define better strategies to respond immediately in case of incidents and thus to save more lives with less effort. In order to make a fair comparison, the average number of interventions per hour was established as a baseline to compare the LSTM model results.

The following sections of this paper are structured as follows: in Section II the state of the art is briefly presented; in Section III the way that data were acquired and encoded is explained; in Section IV feature and model selection are exhibited; in Section V prediction results are given and a discussion highlighting results is provided; in Section VI concluding remarks and future works are given.

II. STATE OF ART

Researches to forecast the number of interventions, response speed, materials and engine used by fire departments are scarce in the literature. However, there are machine learning techniques applied to forecast crime, traffic incidents, traffic flow and speed, occupational accidents, and others. In [5] authors presented a proposal to forecast occupational accidents using Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP) optimized by Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), where the aim was to predict the accident outcomes such as injury and property damage using occupational accident data.

As another example, [6] worked with a deep learning approach, using LSTM neural networks to predict the risk of the traffic accident, taking into account weather variables, periodical patterns and the spatial distribution patterns of traffic accidents; the traffic accident data were discretized in space and time to train the deep model; then, the model was provided with recent historical data to eventually obtain the real-time traffic accident risk prediction.

Gerber [7] presented a study on the use of spatio-temporally tagged tweets for crime prediction using Kernel Density Estimation (KDE) differing from the standard hot-spot models. The author collected crime data for a specified period from the Chicago Police Department and tweets tagged with geographical positioning system coordinates

in the limits of Chicago and Illinois cities. Subsequently, author quantified the crime prediction gains using twitter-derived information to compare to a standard KDE approach achieving improvements for 19 of the 25 crime types.

In [8] authors developed a LSTM NN to capture the nonlinear traffic dynamic to catch the long-term temporal dependency for predicting short-term travel speed. The data were collected for a month from two different locations in Beijing and were divided in twenty five days for training and five days for testing. The performance of the proposed LSTM model was compared with others RNN and artificial intelligence methods. Results for LSTM NN were superior to other methods in most of the tests proving to be an effective approach for short-term travel speed prediction.

[9] proposed a LSTM regression model to predict 24-hours traffic counts data by collecting and labelling 24-hours traffic data of some districts in Texas, constructing the LSTM and a logistic regression model to compare results, and finally analyzing the potential resource allocation patterns according to the obtained results. Results proved that the LSTM regression model is more powerful than logistic regression and that the mean square error of logistic regression is about 5 times higher than LSTM. Conclusion remarks for resource allocation strategy, were that using LSTM would provide more accurate results to decide the distribution of traffic resources.

Further, [10] developed a data-driven toolchain to forecast the likelihood of vehicular incidents in given time and location using incident data from the Nashville Fire Department in USA, considering weather and road type information as added variables; authors combined a similarity based agglomerative clustering to categorize incidents with similar characteristics, a survival analysis to learn the probability of incidents per cluster and a bayesian network inference technique to map the clusters to the spatial locations.

III. DATA ACQUISITION AND ENCODING

The fire and rescue department, SDIS 25, in the region Doubs-France has provided us with a list of interventions collected from 2012 to 2017 containing information on a total number of 195,628 interventions. This data file contains: the identifier code of an intervention, date and time of the beginning and end of an intervention, date and time in which the first engine arrives, the community in which the intervention take place, the classification of the incident, response time and duration of the intervention.

In this section, it will be explained how the data were collected and prepared before being selected and used as input in the prediction phase.

A. Data Acquisition

From the collected list of interventions were extracted from each individual intervention the date and time of when it happened, in order to detect tendencies correlated with these parameters. For instance, the number of car accidents increases on Saturday nights because young people tend to drink and drive during this period of time.

Other factors considered in the occurrence of incidents are the weather conditions, that affect significantly the number of road accidents, fires and casualties; traffic hours, height of the main rivers in the Doubs, epidemiological data, academic vacations, holidays, moonrise, moonset and moon phases. All of this in order to predict the number of interventions that will occur in the following hour or in the following three hours. Therefore, at the beginning, a dictionary was created in which extracted data from the fire department were save and the supplementary data were imported from other sources together. The process is explained as follows:

- The dictionary is initialized containing keys ranging from "01/01/2012 00:00:00" until "31/12/2017 23:00:00" of the form "YYYYMMDDhhmmss". The keys are generated by blocks of 1 hour.
- The following weather-related data reported by "Meteo France" [11] was imported from three stations located in Dijon-Longvic, Bale-Mulhouse, and Nancy-Ochey: temperature, pressure, pressure variation each 1h, barometric trend types, total cloudiness, humidity, dew point, last hour precipitation, last three hours precipitation, average wind speed for every 10 minutes, average wind direction for every 10 minutes, bursts over a period, horizontal visibility, and finally the current time. However, the data were not complete, for this reason, it was necessary to apply a linear interpolation to fill the blanks. Finally, the meteorological data were added to the dictionary.
- It was also introduced various temporal information as: hour, day, day in the week, day in the year, month and year.
- The height of twelve rivers, which are the most representative of the Doubs department were also considered. The data were collected from "Hydro" [12], then the dictionary was filled with the average of the readings closest to the time of the block considered, the standard deviation of variation of the water height on this block, the number of readings during this block, the maximum height occurred during this block of time with the alert 1 (true) if the height of the river exceeds a limit established as a flood alert or 0 (false) if not.
- From the list of interventions given by the fire brigade department, the interventions were classified according to the date of occurrence, grouping them by a period of one hour to add them to the dictionary.
- Holidays were considered as a binary variable initialized with 0 (false), that would be 1 (true) for any 1h block within an academic holiday period. Also, the starting and ending days of vacations were considered as a binary variable where it is 1 (true) for the days corresponding to the beginning and end of holiday periods and 0 (false) if not.
- Public holidays were added (1 or 0, for true or false), as well as a second key that is set to 1 the days before public holidays, for the hours ranging from 3:00 pm to 11:00 pm (otherwise 0).

- Information related to the "Bison Futé" [13] were included, which is a system put in place in France to communicate to motorists all the recommendations of public authorities regarding traffic, traffic jams, bad weather, accidents, advice, etc. It classifies the days at risk according to several colors: green = fluid traffic, orange = dense traffic, red = difficult traffic, black = to avoid because of traffic jams and slow traffic. These information were integrated through two additional keys related to the departure and the return. They are 0, 1, 2 or 3, depending on whether the traffic forecasts correspond to green, orange, red or black.
- Weekly epidemiological information organized by each given hour and related to the incidence of chicken-pox, influenza and acute diarrhea, were incorporated to the dictionary. The data were collected from the "Sentinelles" network [14].
- Finally, a boolean variable to know if it is a day (0) or night (1) for each given hour was added. Moreover, another boolean variable was included to recognize if the moon had already risen. This procedure considered the given hour plus 30 minutes, and what the moon phase was.

B. Data Encoding

Often, the initial data do not have the format feature/value. For this reason, it is necessary to extract what could be potentially useful features to convert them in the format that the model needs as input. This process is called feature extraction [15].

To complete this task, "StandardScaler" from Scikit-learn library [16], which is a method that standardizes features by rescaling the distribution of values to zero mean and unit variance, was used; the variables standardized were: year, hour, wind direction, humidity, nebulosity, moon phase, dew point, precipitations, rafales, temperature, visibility, wind speed, chickenpox statistics, influenza statistics, acute diarrhea statistics, rivers height variables except by the alert variable, etc. Also, from Pandas library [17], the "Get Dummies" (OneHotEncoder) method, that converts categorical variables into dummy/indicator features was used; for instance: bison futé variables, time variables such as day, day of the week, day of the year and month, holiday indicator, night indicator, barometric trend, river height alert variable, etc. Finally, the target feature "nbInterventions" was not standardized, after several tests, it was decided to keep the original values.

IV. FEATURE AND MODEL SELECTION

A. Feature Selection

Usually, using every available feature in the learning data set seems preferable, since it is believed that using as much information would build a better model. However, there are two main reasons why the number of considered features should be restricted. First, it is possible that unimportant features could establish correlations between features and the target that arise just by chance and would not correctly model the problem, leading to poor generalization or adding

redundant information. And second, a large number of features could greatly increase the computation time without any improvement [15].

As a result, working with a smaller set of features may conduct to better results. So, for this work, the "Mutual Info Regression" method from the open source library Scikit-learn [16] was used to select features. Mutual info regression is a nonparametric method to evaluate the mutual correspondence for a continuous target variable based on entropy estimation from k-nearest neighbors distances. Two random features are chosen to calculate their mutual information, the result is a nonnegative value which measures the dependency between the features. When two random variables are independent the output is zero, but higher outputs mean higher dependency.

To get mutual information regression scores, 800 neighbors were used; then, scores were scaled between 0 and 1. Finally, the threshold was established as 0.01, all features with scores higher than 0.01 are considered as input to our model. Hence, from a total of 830 features extracted, 56 were taken to train our model.

B. Model Selection

Another important step is the selection of the model parameters, known as hyperparameters. In this investigation, the hyperparameters considered were the number of layers, the number of units, epochs, batch size and learning rate to build our LSTM model. The models were developed with Keras library [18].

First, after encoding and selecting the features, data were split into three data sets, the training set (2011-2015), the validation set (2016) and the testing set (2017). The first and second set are used to build a model that will predict the number of interventions for 1h and 3h blocks and the latter is used to verify the accuracy of these predictions.

After several tests, the results improved by establishing a constant number of units in the first and second layer of our LSTM architecture, the other hyperparameters were chosen within a range as described in Table I. Then, a random search is performed with 50 iterations searching for the best hyperparameters. The number of units for the third layer, batch size and learning rate change in each iteration randomly to discover which combination of values gives a better result. The maximum number of epochs is 5000, but an *early stopping* method from Keras [18] was used, i.e., if after 15 epochs the loss function of validation set did not improve, the LSTM NN stopped.

The best 10 results were saved to be the starting population for our genetic algorithm, which is different from the original one better detailed in [19]. The GA was also executed 50 more times; this process is implemented in order to tune the hyperparameters obtained before. The steps for the developed GA (Fig. 2) are described bellow:

1) A population of size ten is defined, each individual has the following parameters: number of units in the 1st, 2nd and 3rd layer, batch size and learning rate.

TABLE I: Hyperparameters Settings

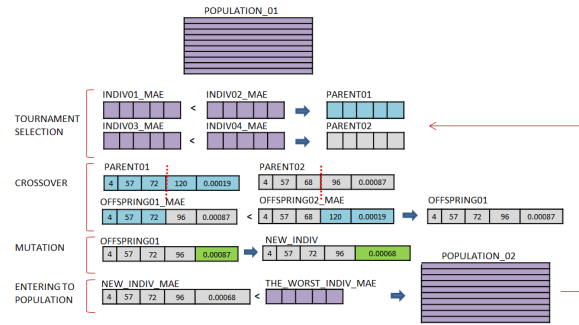| Hyperparameter	Values
Units first layer	4
Units second layer	57
Units third layer	[50, 100]
Epochs	[60, 5000]
Batch size	[55, 160]
Learning rate	[0.000001, 0.001]

Fig. 2: Genetic algorithm design

2) Two tournaments are made and in each one two individuals are selected randomly and their Mean Absolute Error (MAE) is compared, the individual with the best MAE wins the tournament.
3) The two winners are combined to generate two offspring.
4) The offspring with the best MAE is mutated. An individual position is selected randomly, if the position belongs to the number of units or batch size, the value will be mutated by adding or subtracting 1; if the position belongs to the learning rate, the value will be multiplied by a number randomly generated between 0 and 1.5.
5) If the new mutated individual is better than the worst individual in the population, then the latter will be replaced by the former and a new iteration starts from step 2.

V. PREDICTION RESULTS AND DISCUSSION

A. Prediction Results

In this section results are presented and analyzed. The metrics used to evaluate the results are the Root Mean Square Error (RMSE), the MAE as principal and AC1E as the accuracy with a maximum error margin 1. In order to compare with some reference values, this paper considers as baseline the average value calculated for each block of hour, which achieved scores as RMSE 2.4724, MAE 1.8013 and AC1E 53.93% for 1h prediction and RMSE 4.9259, MAE 3.6702 and AC1E 28.28% for 3h prediction. Table II presents results of the LSTM model trying to predict the number of interventions for the next one and three hours using different values 3, 12 and 24 hours of time steps with the best performance marked in bold. Fig. 3 exhibits results for 200 samples to the best 3 hours as time steps trying to predict one hour in the future and Fig. 4 the exact number of

TABLE II: Prediction results on test data 2017

Time steps	Predicting next 1h			Predicting next 3h		
	RMSE	MAE	AC1E(%)	RMSE	MAE	AC1E(%)
3h	**2.30**	**1.69**	**55.41**	4.51	3.37	30.82
12h	2.29	1.70	55.20	4.29	3.23	30.51
24h	2.31	1.70	54.80	**4.22**	**3.19**	**31.26**

predictions with 0 to max 14 errors. The LSTM architecture for this case is composed of 3 LSTM layers with 4, 57 and 77 units respectively and 1 Dense layer with one neuron as output. Similarly, Fig. 5 exhibits the prediction results for 200 samples to the best 24 hours of time steps for the next three hours in the future and Fig. 6 the exactly number of predictions with 0 to max 25 errors. For this case the LSTM architecture consists of 3 LSTM layers with 4, 57 and 86 units respectively and 1 Dense layer with one neuron as output. Note that in Figs. 5 and 6, 200 samples represent 600 ones from the previous case due to the fact that samples are grouped by 3 hours instead of just 1. Also, results were rounded to integer values, i.e., considering that the LSTM model predicts real values (e.g., 2.31 interventions) results were transformed to integer ones (e.g., 2 interventions) to be consistent with real applications.

B. Discussion

The aim of this work was to predict the number of interventions using data from 2012-2015 for training, 2016 for validation and 2017 for testing, and different time step configurations for the next one and three hours in the future. As presented in Table II, for the first case, the model with 3 time steps outperforms the others regarding to MAE metric. As one can see in Fig. 3 the LSTM model is trying to find a fairly acceptable sequence pattern for future interventions, that is, when the number of interventions increases the model tends to increase smoothly and in the same way when there is a valley. Additionally, the analysis of AC1E shows that this model is able to predict more accurately, from a total of 8760 interventions, 1845 are exactly predicted which represents 21.06% of accuracy, but if it is considered an error margin 1 the number of correct predictions increases to 4854 (55.41%), which means that considering such tolerance, the model continues being feasible in real applications. For the second case, the result using 24 time steps is better than 3 and 12 respecting to all metrics, from a total of 2920 interventions, 329 are exactly predicted (11.26%) and with a margin error 1 increases to 913 (31.26%). Contrary to the previous case, this model remains more or less continuous even in spite of peaks and valleys it does not try to increase or decrease the sequence pattern value. This behaviour could produce higher RMSE and MAE in contrast to the first case.

Another interesting fact to mention is that the number of intervention per year is increasing, which is the expected due to population growth. However, this increment is not linear and the year 2017 was exceptionally larger than previous ones as informed by the SDIS 25 fire Department, making predictions more difficult even for our LSTM model.

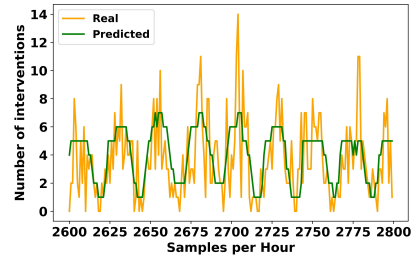Finally, the best 60 models generated for the next one hour

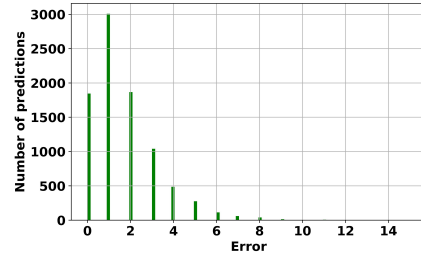Fig. 3: Predictions for 1h future

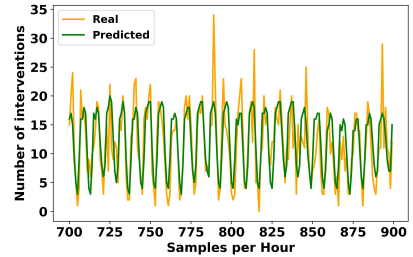Fig. 4: Exact predictions for 1h
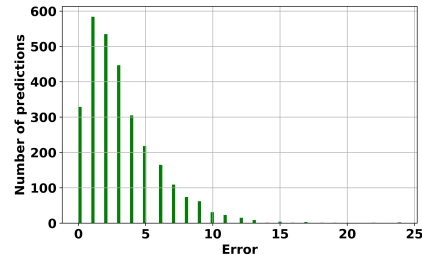

Fig. 5: Predictions for 3h future

Fig. 6: Exact predictions for 3h

prediction were taken in order to analyze and understand the tendency of our hyperparameters. For this task, it was used the fANOVA framework developed by Hutter et al. [20], which is a tool that uses different hyperparameter settings to fit a random forest that captures the relationship between hyperparameters and performance and then evaluates their influence. During the assessing, the individual importance was calculated with 200 trees and the inputted data were: number of units in the third layer (N3L), epochs (EP), batch size (BS), learning rate (LR) as features and MAE metric as the response. Results in Fig. 7 showed that the most important hyperparameter was the number of epochs with 52.06%, which might be due to the fact that the number
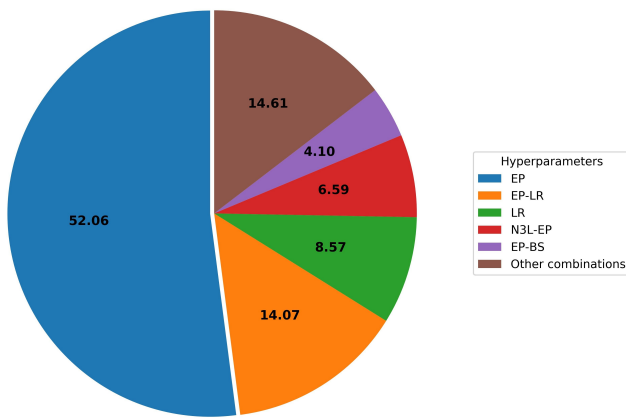
Fig. 7: Individual hyperparameter importance

was dynamic and well-treated with Keras's early stopping function during the training. The second one is the pairwise epochs and learning rate with 14.07% of importance followed by the learning rate by itself with 8.57%, which means that with a fine-tuned step size we can achieve better results keeping third layer values between 70 and 80, batch size between 80 and 90, and more than 200 epochs according to the fANOVA results for each hyperparameter.

VI. CONCLUSION

The ability to predict the number of interventions at a given time in the future could help fire departments around the world to deal more efficiently with forthcoming incidents and to improve the management of human and mobile resources. This paper presents a long short-term memory neural network to predict the number of interventions of firefighters in the region of Doubs-France to the next one and three hours. To validate the performance of the developed LSTM model, 6 years of interventions data from the departmental fire and rescue SDIS 25 in the region Doubs-France were collected. Results demonstrated that the LSTM model is trying to find a sequence pattern to predict the number of interventions, which is suitable for practical purposes to anticipate the management of resources to improve response time. We do recognize that these results have to be improved in a more accurate way to be used as first decision making approach by fire departments.

For future work, we will test different machine learning methods, better explore LSTM architectures and combinations with other neural networks models (e.g., convolutional neural networks) and search for different optimization methods to improve results using intelligent techniques. Other attractive directions are to analyze the features with other methods for feature selection (e.g., F-test, Principal Component Analysis - PCA), find a way to model the data that allows to learn the increment of the number of interventions over the years, and increase the number of time steps. Moreover, we will develop techniques capable of predicting the type of intervention and the location to complement and work as toolchain with this research.

REFERENCES

[1] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer Berlin Heidelberg, 2012. [Online]. Available: https://doi.org/10.1007/978-3-642-24797-2

[2] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, nov 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[3] A. Géron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow*, 1st ed. 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, 7 2017, vol. 1.

[4] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, oct 2017. [Online]. Available: https://doi.org/10.1109/tnnls.2016.2582924

[5] S. Sarkar, S. Vinay, R. Raj, J. Maiti, and P. Mitra, "Application of optimized machine learning techniques for prediction of occupational accidents," *Computers & Operations Research*, mar 2018. [Online]. Available: https://doi.org/10.1016/j.cor.2018.02.021

[6] H. Ren, Y. Song, J. Wang, Y. Hu, and J. Lei, "A Deep Learning Approach to the Citywide Traffic Accident Risk Prediction," *arXiv e-prints*, Oct. 2017. [Online]. Available: https://ui.adsabs.harvard.edu/\#abs/2017arXiv171009543R

[7] M. S. Gerber, "Predicting crime using twitter and kernel density estimation," *Decision Support Systems*, vol. 61, pp. 115–125, may 2014. [Online]. Available: https://doi.org/10.1016/j.dss.2014.02.003

[8] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, may 2015. [Online]. Available: https://doi.org/10.1016/j.trc.2015.03.014

[9] X. Du, H. Zhang, H. V. Nguyen, and Z. Han, "Stacked LSTM deep learning model for traffic prediction in vehicle-to-vehicle communication," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*. IEEE, sep 2017. [Online]. Available: https://doi.org/10.1109/vtcfall.2017.8288312

[10] G. Pettet, S. Nannapaneni, B. Stadnick, A. Dubey, and G. Biswas, "Incident analysis and prediction using clustering and bayesian network," in *2017 IEEE SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI*. IEEE, aug 2017. [Online]. Available: https://doi.org/10.1109/uic-atc.2017.8397587

[11] [Online]. Available: http://www.meteofrance.com/accueil

[12] [Online]. Available: http://www.hydro.eaufrance.fr/

[13] [Online]. Available: https://www.bison-fute.gouv.fr/

[14] [Online]. Available: https://www.sentiweb.fr/?page=table

[15] R. Garreta and G. Moncecchi, *Learning scikit-learn: Machine Learning in Python*. Packt Publishing, 2013.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[17] W. McKinney, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, S. van der Walt and J. Millman, Eds., 2010, pp. 51 – 56.

[18] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[19] K. Sastry, D. Goldberg, and G. Kendall, "Genetic algorithms," in *Search Methodologies*. Springer US, 2005, pp. 97–125. [Online]. Available: https://doi.org/10.1007/0-387-28356-0_4

[20] F. Hutter, H. Hoos, and K. Leyton-Brown, "An efficient approach for assessing hyperparameter importance," in *Proceedings of International Conference on Machine Learning 2014 (ICML 2014)*, Jun. 2014, p. 754–762.