Data-driven models in machine learning for crime prediction

Zbigniew M. Wawrzyniak
Faculty of Electronics and Information Technology
Warsaw University of Technology
00-665 Warsaw, Poland
ORCID: 0000-0003-0052-4114

Zbigniew Szymański
Faculty of Electronics and Information Technology
Warsaw University of Technology
00-665 Warsaw, Poland
ORCID: 0000-0002-4789-414X

Stanisław Jankowski
Faculty of Electronics and Information Technology
Warsaw University of Technology
00-665 Warsaw, Poland
sjankows@elektron.elka.pw.edu.pl

Radosław Pytlak
Faculty of Mathematics and Information Science
Warsaw University of Technology
00-665 Warsaw, Poland
ORCID: 0000-0001-6181-804X

Grzegorz Borowik
Faculty of Electronics and Information Technology
Warsaw University of Technology
00-665 Warsaw, Poland
ORCID: 0000-0003-4148-4817

Eliza Szczechla
Scott Tiger S.A.
01-692 Warsaw, Poland
eliza.szczechla@ tiger.com.pl

Paweł Michalak
Scott Tiger S.A.
01-692 Warsaw, Poland
pawel.michalak@tiger.com.pl

***Abstract*—Prediction of future events over time is associated with a sequence of time series observational samples and other exogenous data. Different approaches connected with statistical learning techniques result in predictive data-based models. The paper presents an attempt to develop techniques for predictive data-based modeling based on machine learning data-driven approaches. To reach a good level of prediction we use a deep learning architecture based on artificial neural network (ANN). The neural network (NN) structure for crime prediction and the appropriate inputs for crime prediction is performed through: Gram-Schmidt orthogonalization (GS) for the selection of network inputs and virtual leave-one-out test (VLOO) for the selection of the optimal number of hidden neurons. Spatiotemporal distribution of the hot-spots is conducted and a methodology is developed for short-term crime forecasting using the long short-term memory (LSTM) recurrent neural networks (RNN) and convolutional neural networks (CNN).**

Keywords—data-based model, statistical learning, crime prediction, forecasting, neural network, LSTM

I. Introduction

Prediction of future events over time is associated with a sequence of time series (TS) observational data (regularly or irregularly spaced) and other exogenous variables affecting crime. Knowledge extracted from available open sources and structured data from police records can be utilized to understand criminal behavior better and to forecast future crime events to efficiently prevent crime basing on available data and analytic models from criminology, sociology and other areas of knowledge and experience of law enforcement agencies (LEA).

As a result of the study will be applied to real-world crime prediction problems, they offer support for policy and decision planning making it ready for deployment by the of Police and LEA agencies.

In our work, we study a crime event data set (observational data) obtained from undisclosed police records as part of the crime forecasting project.

The paper shows an attempt to develop techniques for predictive data-based modeling based on machine learning data-driven approaches. To reach good level of prediction we use a deep learning architecture based on ANN.

The selection of an optimal NN and appropriate inputs for crime prediction are considered is performed through: GS orthogonalization for the selection of network inputs and virtual leave-one-out test (VLOO) for the selection of the optimal number of hidden neurons. Spatiotemporal distribution of the hot-spots is conducted and a methodology is developed for short-term forecasting using the long short-term memory (LSTM) recurrent neural networks (RNN) and convolutional neural network (CNN).

II. Models

A. Data-driven models

Different models are used for social processes description [1, 2]. However, most models commonly used in the field is a model-driven approach (analytical model). Data representing the underlying ground truth are being collected and stored in the process of modeling. In real applications, a data set can contain observations that are strongly influenced by external factors unknown to researchers, that may cause a set of samples to significantly diverge from the assumed distribution. One approach is to create a black-box model using the data collected and study the results to produce an updated analytical solution. Depending on the simulation or modeling aims this approach can be treated as a data-based decision model or a data-driven process model.

As the model-based concept is a white box one, it uses deep analytical (domain-base) knowledge about the underlying processes with exogenous and endogenous parameters [3]. The other, as opposed to it, is a black-box concept (model-free), when a data-driven model is fit on inputs and outputs. Depending on the discrete (equally

The work was funded by the grant no. DOB-BIO7/05 /02/2015 of Polish National Center for Research and Development (NCBiR).

978-1-5386-7834-3/18/$31.00 ©2018 European Union

spaced sampling or stochastically dependent on time) or quasi-continuous data they can be categorized into two problems: classification or regression. However, even continuous models of analytical origin for can be sampled due to simplicity in discrete machine simulation. Arguments coming from state of the art technologies about data collection possibilities by various types of sensors also support this approach.

Practically we can treat such observational data as regularly spaced (after some aggregation on irregularly spaced data with missing values [4]) or stochastically generated ones (for information uploaded to data collectors at unequal time intervals) for further supervised or unsupervised categorization. The crime TS is a nonstationary process whose unconditional joint probability distribution change when shifted in time or has random and episodic variations. It is in the nature of a TS of crime data collected irregularly with significant gaps of missing data or collected with irregular sampling intervals or both. Therefore, the methods of harmonic analysis were not applied directly as to a periodic behavior of TS [4].

A combination of the data-driven and model-driven approach to practical problem description and control in a hybrid model, especially with social physics [5, 6] and mechanics [7] when observational data are available, results in almost comprehensive models applicable for active controlling and forecasting and planning (decisions and prediction). Finally, we can observe that gaining knowledge, making predictions and decisions, or constructing the most suitable models from a set of data require a statistical learning theory framework based on multivariable time-series data with estimation of the uncertainty of each model. Generally, in social processes as a crime [8, 9, 10], the characteristics of past events influence the current ones. This phenomenon can be naturally represented in a Long Short-Term Memory network (LSTM) [11, 12]. Neural network learning with LSTM is capable of predicting some event time samples (steps) ahead of the input [13]. The data-based black box models include different object rules in the form of linear and nonlinear autoregressive models. Results revealed that calibrated optimized nonlinear piecewise model was good enough to capture the dynamics of this kind of data, for example, to the model some event spikes and anomalies but the limitations is the data dimensionality and direct access to available data.

B. The data-dependent problem in crime modeling

As the dynamics of crime process depend on many complex factors and social phenomena, for the public safety organizations are the end-user of sophisticated technological services, an accurate real-time crime prediction seems to be a fundamental issue in the present-day world but remains a challenging problem for the scientists and practitioners. Compared to many predictable types of events, crime is rather sparse in observed data over multiple time periods at different spatiotemporal scales and factorial features presenting different patterns in these dimensions of space, time, and factors (explanatory variables).

The study reported here is an attempt to develop techniques for predictive data-based modeling based on machine learning data-driven approaches by ANN. Data characteristics of appropriate inputs for crime prediction influence a selection of optimal NN architecture: a number of network inputs and a number of hidden neurons.

III. NN Methodology

In this section, the problem of optimization of a NN is described, as well as the LSTM architecture for the specific question of irregularly spaced observation with a slight variance periodic component.

The ANN networks are data-driven flexible models (so-called model-free) that are capable of approximating a large class of nonlinear TS problems to any desired level of accuracy. In TS forecasting, multilayer perceptron (MLP) is used widely in the forecasting problems, the number of inputs and number of neurons in the hidden layer is flexible depending on the number of acyclic internal rules.

The selection of optimal NN and the appropriate inputs for crime prediction are considered by using: GS orthogonalization for the selection of network inputs and VLOO for the selection of the optimal number of hidden neurons.

Based on ANN architecture the LSTM models were designed for the specified number of crime counts for subsequent days and weeks. The TS representing daily crime counts in the preceding periods were designed for the selected regions.

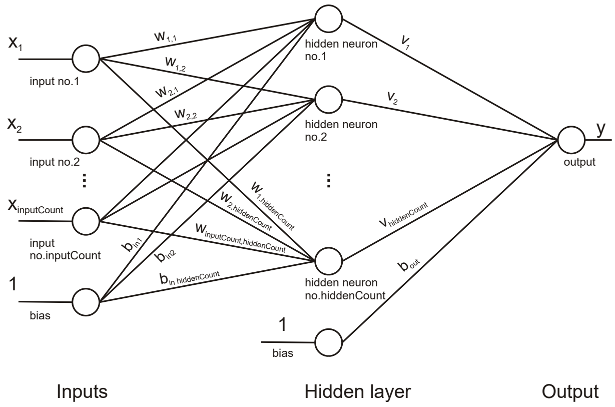

Fig. 1. Neural network structure.

A. Optimal architecture of ANN

The goal of this section is the selection of an optimal NN for the crime event prediction. It is a nonlinear extension of linear ARIMA models for TS prediction. The NN architecture is shown in Fig. 1 and is a MLP with one output neuron, one hidden layer, and the input layer. The design task is the appropriate selection of network inputs – the most relevant delayed TS values and the number of hidden neurons performing the nonlinear prediction representation. The active functions of hidden neurons are a hyperbolic tangent, and the output neuron is linear. The NN output is

$$y^n = g(\mathbf{x}, \mathbf{w}) = \sum_{j=1}^{J} W_j \tanh(\sum_{i=1}^{I} w_{ji} x_i^n), \qquad (1)$$

where: i = 1, …, I – index of network inputs, j = 1, …, J – index of hidden neurons, n =1 ,…, N – index of training data set.

The training set for the application of a NN for TS prediction is defined as pairs of the input vector composed of past values $\mathbf{x} = (x(t–\Delta t), x(t–2\Delta t),…, x(t–m\Delta t))$ and the future value $d = y = x(t+\Delta t)$. The NN error r_n is equal to the difference between the desired d and the calculated value y: $r_n = d^n - y^n$. The mean square error function TMSE for entire training set is

$$TMSE = \sum_{n=1}^{N} E^n = -\frac{1}{2}\sum_{n=1}^{N}\left[d^n - y^n\right]^2 = -\frac{1}{2}\sum_{n=1}^{N}\left[d^n - \sum_{j=1}^{J} W_j \tanh\left(\sum_{i=1}^{I} w_{ji}\, x_i^n\right)\right]^2, \qquad (2)$$

E is the function of network parameters $\{\mathbf{w}\}$ and training set inputs $\{\mathbf{x}^n, d^n\}$.

1) Data Structure for NN

The structure of the input data fed to the NN is explained in Tab. 1. The input of the NN is a vector of k+1 consecutive values from the TS. The value of parameter k depends on the implemented model structure. The model variable ranking procedure enables selection of most relevant NN inputs (some NN input columns for other variables in Tab. 1 are removed).

TABLE I. STRUCTURE OF LEARNING DATA

NN variables				
inputs				**output**
$y(0)$	$y(1)$	…	$y(k)$	$y(k+1)$
$y(1)$	$y(2)$	…	$y(k+1)$	$y(k+2)$
…	…	…	…	…
$y(i)$	$y(i+1)$	$y(2)$	$y(k+i)$	$y(k+i+1)$

2) Model variable ranking

Orthogonalization procedures enable our examination of the influence of every input feature on the output vector [14]. The presented method uses simple ranking formula and the well-known GS orthogonalization procedure for pointing out the model most salient variables [15]. N input-output pairs (measurements of the output of the process to be modeled, and of the candidate features) is available. We denote by:

- Q, N – a number of candidate features and measurements of the process to be modeled, respectively,
- $\mathbf{x}^i = [x^i_1, x^i_2, ...x^i_N,]$ – the vector of the i-th feature values of N measurements,
- $\mathbf{y}_p$ – the N-dimensional vector of the classifier target values.

We consider the N by Q matrix X = $[\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^Q]$. The ranking procedure starts with calculating a correlation coefficient $\cos^2(\mathbf{x}^k, \mathbf{y}_p)=(\mathbf{x}^k\,\mathbf{y}_p)^2/(\|\mathbf{x}^k\|^2\ \|\mathbf{y}_p\|^2)$. The larger it is, the better the k-th feature vector explains the $\mathbf{y}_p$ variation.

As the first base vector, we pick the one with the largest value of the correlation coefficient. All the remaining candidate features and the output vector are projected onto the null subspace (of dimension N–1) of the selected feature. Next, we calculate the correlation coefficient for the projected vectors and again pick the one with the largest value of this quantity. The remaining feature vectors are projected onto the null subspace of the first two ranked vectors by the classical GS orthogonalization. This procedure is continued until all the vectors $\mathbf{x}^k$ are ranked.

To reject the irrelevant inputs, we compare its rank value correlation coefficient with the rank of a random probe. The remaining features are considered relevant to the model. During the training the NN, the GS variable selection procedure revealed the most relevant variables.

3) Selection of neural predictor by VLOO

The selection of the NN architecture is usually based on the minimum mean square errors on training and test sets [16, 17]. However, the NN tends to overfitting of a training data set if the number of network parameters is too large. This effect is not easy to discover. The novel idea to avoid the overfitting is to test the influence of training data set on the obtained predictor. The model of the best-estimated generalization – efficient for new data – should be characterized by the equal influence measure for all training data set.

The leave-one-out cross-validation is the method for estimation of the influence of each training data. However, it is computationally inefficient [18]. Therefore the estimation of individual data influence is estimated in this paper by using the VLOO score [19, 20, 21]. The nonlinear model in the form of MLP is locally linearized. In the vicinity of the error function minimum $\mathbf{w}^*$ a nonlinear model g($\mathbf{x},\mathbf{w}^*$) can be approximated by using Taylor series expansion:

$$g(\mathbf{x},\mathbf{w}) = g(\mathbf{x},\mathbf{w}^*) + \mathbf{Z}(\mathbf{w}\text{-}\mathbf{w}^*), \qquad (3)$$

$\mathbf{Z}(N, q)$ – Jacobian matrix of the nonlinear NN of size (N,q), $N > q$, where: N–the number of training examples, q–the number of the model parameters**.** The local linear approximation of the NN solution has the least-squares form:

$$\mathbf{w}_{LS} = \mathbf{w}^* + (\mathbf{Z}^T\mathbf{Z})^{-1}\,\mathbf{Z}^T\,[y_p - g(\mathbf{x}, \mathbf{w}^*)]. \qquad (4)$$

The estimated effect on the network weights of withdrawing one example of the training set without relearning

$$\mathbf{w}_{LS}^{(-k)} = \mathbf{w}^* + (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{z}^k\,\frac{r_k}{1-h_{kk}}, \qquad (5)$$

where residual r_k is:

$$r_k = y_p^k - g(\mathbf{x}^k,\mathbf{w}). \qquad (6)$$

By introducing **H** matrix of an orthogonal projection of the Z matrix onto solution subspace, the effect of withdrawing k-th example from the training set can be estimated:

$$\mathbf{H} = \mathbf{Z}\,(\mathbf{Z}^T\mathbf{Z})^{-1}\,\mathbf{Z}^T. \qquad (7)$$

The diagonal elements h_{kk} of **H** matrix are leverages – k-th components of orthogonal projections. The leverages are equal:

$$h_{kk} = \mathbf{z}^{kT}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{z}^k = \sum_{l=1}^{q}\sum_{j=1}^{q} Z_{kl}Z_{lj}\,(\mathbf{Z}^T\mathbf{Z})^{-1}_{lj}. \qquad (8)$$

The leverages can be calculated by singular value decomposition (SVD). The VLOO residual equals:

$$r_k^{(-k)} = \frac{r_k}{1-h_{kk}}. \qquad (9)$$

If all leverages were identical then $h_{kk} = q/N$.

4) Quantitative criteria of model selection

Prediction error of VLOO test E_p is equal:

$$E_p = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(\frac{r_k}{1-h_{kk}}\right)^2} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(r_k^{(-k)}\right)^2}. \qquad (10)$$

This value can be compared to training mean square error *TMSE*:

$$TMSE == \sqrt{\frac{1}{N}\sum_{k=1}^{N} r_k^2}\ , \tag{11}$$

The quantity E_p can characterize models overfitting a subset of training examples.

The relations hold:

$$\forall k: \quad h_{kk} = \frac{q}{N} \Rightarrow E_p = \frac{N}{N-q} TMSE\,, \tag{12}$$

The leverages distribution can be characterized by the quantity μ:

$$\mu = \frac{1}{N}\sum_{k=1}^{N}\sqrt{\frac{N}{q}h_{kk}}\ . \tag{13}$$

In order to select the best NN architecture, an E_p–μ visualization of quality assessment for selected networks could be prepared. The set of examined models comprises NNs with varying inputs and hidden neurons and training was performed for each network's architecture. The initial weight values were randomly selected in order to avoid local minima.

B. *LSTM Neural Networks*

Predictive modeling problem for crime observational TS is a difficult prediction task as the complexity of a sequence depends on the input and exogenous variables. Internal structures of rules and handling sequence dependence can be modeled by RNN used in deep learning.

1) LSTM architecture

The LSTM network, as a type of RNN, can be successfully trained in the process of minimization of output error on a set of training sequences. Having memory blocks connected through layers, LSTM network manages the block's state and output by conditional gates: "output" and "input" ones, as mini-state machines. Other functional "forget gate" conditionally decides about information throwing away from the block (Fig 2).

The NN outputs for the forward pass of an LSTM unit with a forget gate are:

$$\begin{aligned}
f_t &= \sigma_g\left(W_f x_t + U_f x_{t-1} + b_f\right),\\
i_t &= \sigma_g(W_i x_t + U_i x_{t-1} + b_i),\\
o_t &= \sigma_g(W_o x_t + U_o x_{t-1} + b_o),\\
c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_{t-1}(W_c x_t + U_c x_{t-1} + b_c),\\
h_t &= o_t \circ \sigma_{t-1}(c_t).
\end{aligned} \tag{14}$$

where the initial values are: c_0=0 and h_0=0, and the operator ° denotes element-wise product (the *Hadamard* product). The subscript *t* indexes the time step.

We consider a pre-specified window lagged observation as input for maintaining state between data within one batch via smoothing averaging in the moving window. A batch of data is a fixed-sized daily event number from the training data set that defines how many observations to process before updating the weights of the network.

2) Models for crime prediction

The solution leverages a stacked LSTM architecture. Data are fed into the model via two input layers; First input layer comprises information about daily crime counts of the same type in the preceding week along with a weekday indicator. Another input layer carries information about the weekly cycle for the forecasted period, i. e. weekdays for the three days predicted. The first input layer is connected to an LSTM layer with 32 nodes that connect to another LSTM layer with 256 nodes. The second input layer feeds data to a dense layer with one node. The two sections merge into a dense layer containing 256 nodes and via a batch normalization layer and a *ReLU* activation (a rectified linear unit) produce the output layer of 3 nodes (see Fig. 3)

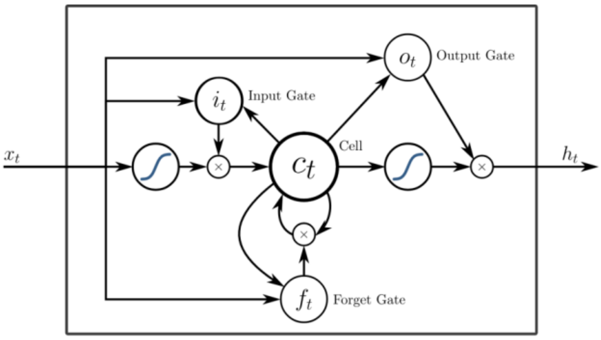

Fig. 2. Schematic functions of conditionally activated gates for a long short-term memory unit at time step (t-1) [22]

The model described in Fig. 3 (called *LSTMday*) predicts daily crime counts for the specified number of subsequent days (e.g., three) based on the TS representing daily crime counts in the preceding week. As weekly seasonality is an essential factor in many crime types studied in our research, the model uses the day of the week as another explanatory variable. It is worth noting that each day is forecasted three times as the time window is shifted: first as day+3, and finally as the subsequent day (day+1).

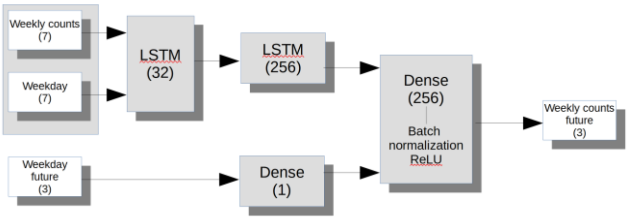

Fig. 3. Stacked long short-term memory architecture for crime counts

The model *LSTMday* predicts daily crime counts of a specified type (e.g., INT) for a selected region. We fit our model to the training set (the part of the chronologically arranged data set) and run the trained model on the remaining part of the data. The predictions were aligned on a common time scale to allow for a comparison of the results for each day forecasted as day+3, day+2, and day+1, as the time window was being shifted.

The LSTM model designed for the other region (called *LSTMweek*) predicts weekly crime count based on the TS representing weekly crime counts of the same type in the preceding ten weeks. The time window is shifted by one day for each sample. The information about the day of the week opening the 7-day cycle is used as an additional explanatory variable.

An interesting idea based preliminary on the analysis of our data is another variation of this model, leveraging the annual cycle (called *LSTMyear*). In this version, we added an explanatory variable representing the weekly count of the same crime type from the corresponding period of the preceding year. It imposes an additional requirement on data availability as the register must extend further back in the past.

IV. Input Data

In our work, we study a crime event data set obtained from undisclosed police records as part of the crime forecasting project PROKRYM. The data sets consist of spatially and temporally referenced observations from multiple areas of two regions (voivodeships) of Poland.

For our work, the data sets have been aggregated by crime type and region. Subsequently, a further grouping was applied and the records were aggregated by time of the event. The granularity of the time grouping varies between models. Due to the high sensitivity of the data, the values representing the aggregated number of crimes have been normalized to the range (0; 1) for visualization in this paper (Results part). We use the first 70% of the records as training data and leave out the remainder as the test set.

In this part of our study presented in the Results section, we adopted a principle that samples processed by one model come from the same region, namely BA and GD, and refer to one particular crime type. In our paper, we focus on short-term predictions for a selected crime type for research areas in the regions mentioned. The data sets used have no spatial component, and the models are estimated separately for each region

The original data set for NN architecture optimization (called *NNarch–opt*) contained 615,961 records describing events from unstructured police reports. The only data fields used in the study were date and type. The data studied in this work cover the period 2008 to the end of 2014. Another studied data set for LSTM models (called *LSTM NN*) spans the period from January 2013 to May 2016. The anonymized records contain information about 12 types of events (crime types) such as robbery, burglary, misdeed, violence, and interventions (INT) not assigned to any of the categories listed.

The original incident count data were aggregated by day (Fig. 4), and the aggregation was performed by summing up the total number of events and the events of individual types. The processed data were analyzed in subsequent stages of presented work for NN optimization (Optimal architecture of ANN part).

Preprocessing of TS for NN modeling involves: removing of the mean value, differentiation, and normalization. In the first stage, the mean value was subtracted from the time sequence shown in Fig. 4.

The stationary TS manifests fluctuations around the mean value, and its autocorrelation function quickly decreases to 0. If the series has positive values of the autocorrelation function for many delay values (i.e. 10 or more), it means differentiation should be performed. Differentiation tends to introduce negative autocorrelations. The TS was differentiated according to formula

$$x_{diff}(n) = x(n) - x(n-1), \qquad (15)$$

The last processing step was data normalization according to formula

$$x_{dif\,f_{norm}}(n) = \frac{x_{diff}(n)}{std(x_{diff})}. \qquad (16)$$

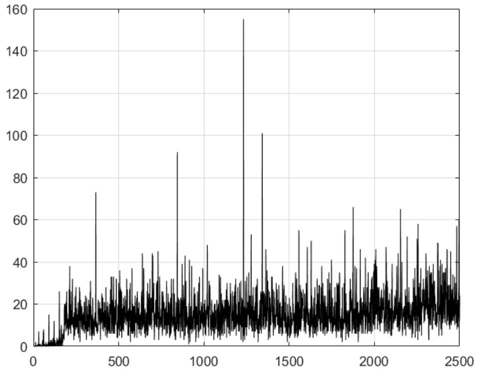

Fig. 4. Original time sequence – the number of events per day (INT, intervention events) for *NNarch–opt* set

V. Results

In this section, the results for numerical experiments for the sets: *NNarch–opt* and *LSTM NN* are described in details based on the optimal architecture for ANN (section 3.1) and the LSTM RNN (section 3.2). For compact naming of NN networks we use the abbreviation of NN(n–m), where n denotes the number of inputs and m denotes the number of hidden neurons of an NN.

A. Optimal architecture of ANN

The set of examined models for the optimal architecture comprises NN with three inputs (selected by the GS procedure) and 3 to 40 hidden neurons. The three selected inputs are delayed by 18, 24 and 28 days.

The best selected NN architecture caused by a quality assessment of selected networks, the E_p–μ space was represented in the prepared 2D plot (Fig. 5). The set of examined models comprises NNs with three inputs and 3 to 40 hidden neurons, and the network training was performed five times for each network architecture. The initial weight values were randomly selected in order to avoid local minima. The preferred models are in the proximity of E_{p} equal to 2.5 and μ equal to 0.95.

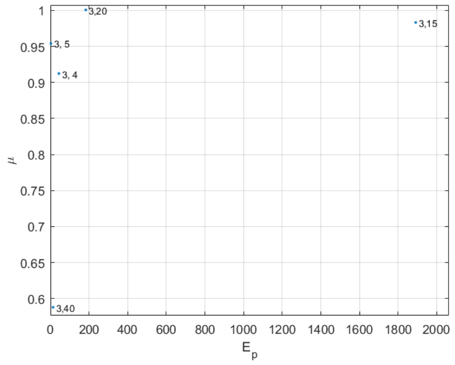

Fig. 5. Prediction error of the virtual leave-one-out test: E_p vs quantity μ (the E_p–μ plot). Each point represents NN(n–m).

Table 2 shows the numerical data used for NN architecture selection. The main criterion was the lowest E_p value. Mean squared errors for the training and test sets are also shown. The selected network NN(3–5) comprises only five hidden neurons and three inputs (selected by the GS procedure).

TABLE II. NN ARCHITECTURE SELECTION RESULTS

Inputs n	Hidden neurons m	E_p	μ	Training mse	Testing mse
3	4	44.01	0.912	0.669	0.633
3	5	2.53	0.954	1.369	0.558
3	10	142651.10	0.961	0.788	1.552
3	15	1891.04	0.983	1.373	28.736
3	20	183.15	1.001	1.233	0.739
3	40	13.86	0.588	1.248	0.645

The experiments for verification of model selection were performed on several subsets of the whole *NNarch–opt* data set containing 200-week samples (short-term forecast). The first 100 samples were used as a training subset, and remaining samples were used for prediction tests. Model learning and prediction tests were performed as one step ahead predictor. Table III shows the decrease of prediction quality when the same NN model is applied to consecutive windows after the window containing the training set.

TABLE III. TEST RESULTS OF THREE NN MODELS ON DIFFERENT SUBSETS

Test subset +shift	NN architecture (n–m)		
	NN (3–5)	NN (3–10)	NN (3–15)
0	0.558	1.552	28.736
+100	0.805	2.574	107.213
+ 200	0.822	1.497	7.323
+ 300	0.726	1.685	67.603
+ 400	1.107	3.196	24.313
+ 500	0.787	3.290	133.967
+ 600	0.841	2.522	29.280
+ 700	0.933	1.937	17.996
+ 800	0.862	2.728	52.871

B. *Predictions with LSTM NN*

For the short time crime prediction from the *LSTM NN* set, we used a stacked LSTM architecture as shown in Fig. 3. The model has been implemented in Keras with Tensorflow back-end. We use the mean squared error (MSE) as an accuracy measure.

In our research, the *LSTMday* model for daily crime count prediction for a selected region BA was used to predict daily crime counts of INT type (intervention) for three subsequent days. As weekly seasonality is an important factor, our LSTM NN models use weekday as the explanatory variable. Figure 6 presents the *LSTM NN* data set used in daily aggregation and calibrated to the interval (0; 1) for the training (70%) and test set (30%) with one day shift for each sample.

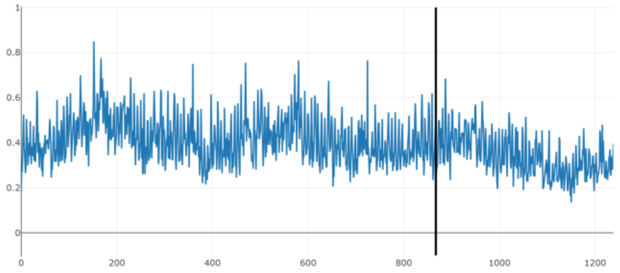

Fig. 6. Daily aggregated crime counts in the time series calibrated to the interval (0;1) with training (70%) and test (30%) subsets with one day shift for each sample

The predictions were aligned on a common time scale to allow for a comparison of the results for each day forecasted as day+3, day+2, and day+1, as the time window was being shifted. We compare the observed values of the daily crime counts for the selected region to the model's predictions for each day as it is processed in three subsequent samples for one-year observation calibrated to the interval (0; 1) (Fig. 7).

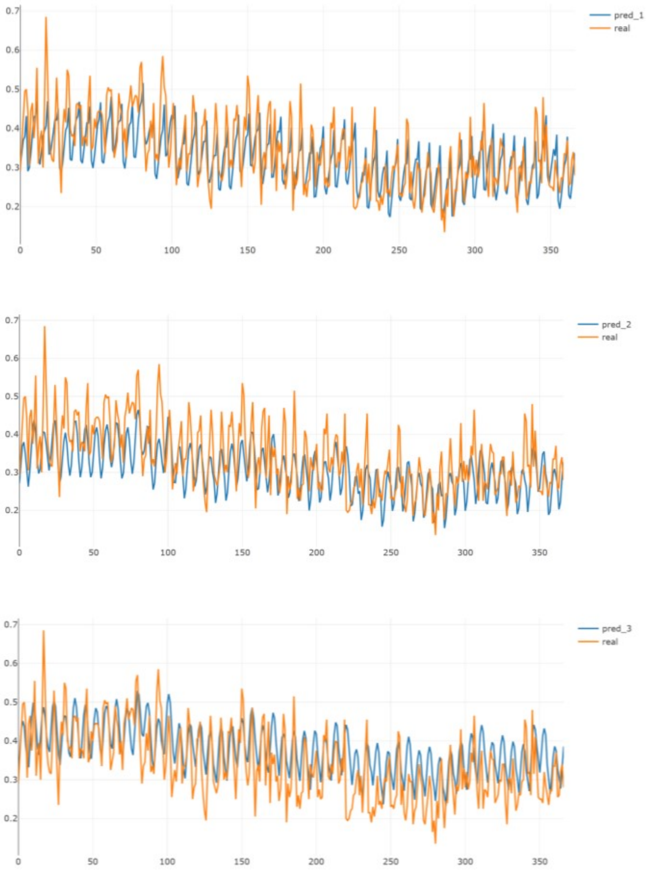

Fig. 7. The one-year prediction aligned on a common time scale per day for three subsequent samples for each day forecasted: day+3 (A), day+2 (B), and day+1 (C)

Next, we compare the observed daily counts to the model's average daily predictions, calculated as an average of predictions for each day processed in three subsequent samples (as day+3, day+2, and day+1) calibrated to the interval (0; 1) (Fig. 8).

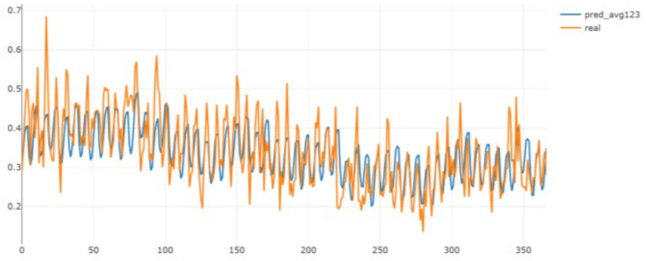

Fig. 8. The one-year prediction aligned on a common time scale per day for averaged subsequent from Fig. 7 (day+3; day+2; and day+1)

The *LSTMweek* model designed for the GD region predicts weekly crime count based on the time series representing weekly crime counts of the same type in the preceding ten weeks. The time window is shifted by one day for each sample. The information about the day of the week opening the 7–day cycle is used as the additional explanatory variable. To compare both models for the regions, the validation error was considered (Fig. 9).

We also studied another variation of this model, i. e. *LSTMyear* leveraging the annual cycle. In this version, we added an explanatory variable representing the weekly count of the same crime type from the corresponding period of the preceding year (influence of other exogenous variables on crimes is presented in, e.g. [23]). Examples of crime data from the region indicate that using historical data (shifted by one year), a lower error could be obtained and therefore a more precise model for forecasting.

It imposes an additional requirement on data availability as the register must extend further back in the past.

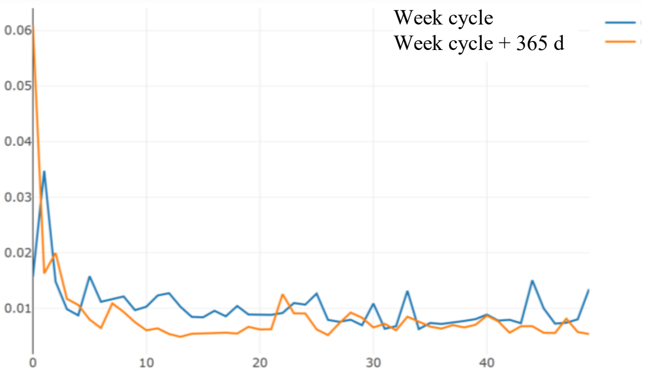

Fig. 9. Model error for the long short-term memory neural network (*LSTMweek*) in the function of epoch during the validation phase for actual and historical data shifted by a year

VI. Conclusions

The neural networks designed for prediction of crime events from unstructured police records could be optimized successfully by using Gram-Schmidt orthogonalization for the selection of network inputs and virtual leave-one-out test for the selection of the optimal number of hidden neurons. Usually, the best neural architecture is designed based on the minimum training set and test set mean square errors shown in the E_p–μ space. Upon these criteria, the best model is NN(3–4) for *NNarch–opt* set. However, upon the virtual leave-one-out criteria, the best is the model NN(3–5).

In order to verify the optimal selection methodology three neural predictors are studied: NN(3–5), NN(3–10) and NN(3–15) as their training set mean square errors are similar. The verification of the virtual leave-one-out model selection shows that the prediction mean-square error of the best model NN(3–5) is much smaller than these errors for the two other models NN(3–10) and NN(3–15) on all test intervals. These results strongly support the efficiency of the presented method of optimal neural predictors selection.

In our paper, we also focus on short-term predictions for a selected crime type by the use of the LSTM models: *LSTMday, LSTMweek* and *LSTMyear*. The data sets used have no spatial component, and the models are estimated separately for each region. The stacked LSTM architecture has been implemented in Keras with Tensorflow back-end for daily and weekly crime count prediction for a selected region. Weekly seasonality and the day of the week were used in our LSTM NN models as explanatory variables.

References

[1] P. Hedström, R. Swedberg and G. Hernes, G. (Eds.), Social mechanisms: An analytical approach to social theory. Cambridge University Press, 1998.

[2] F. Schmalleger, Criminology: A brief introduction. Boston, Prentice Hall, 2011.

[3] N. Ghaffarzadegan, J. Lyneis and G. P. Richardson, “How small system dynamics models can help the public policy process,” System Dynamics Review, vol. 27, no. 1, 2011, pp. 22-44.

[4] S. Dilmaghani and I. C. Henry, P. Soonthornnonda, E. R.Christensen, R. C. Henry, “Harmonic analysis of environmental time series with missing data or irregular sample spacing,” Environmental science & technology, vol. 41, no. 20, 2007, pp.7030-7038.

[5] C. Castellano, S. Fortunato and V. Loreto, “Statistical physics of social dynamics,” Reviews of modern physics, vol. 81, no. 2, 2009, pp. 591.

[6] M. R. D'Orsogna and M. Perc, “Statistical physics of crime: A review,” Physics of life reviews, vol. 2, 2015, pp. 1-21.

[7] J. A. Hołyst, K. Kacperski and F. Schweitzer, “Social impact models of opinion dynamics,” Annual Reviews Of Computational Physics, vol. IX, 2001, pp. 253-273.

[8] R. Akers, R. Social learning and social structure: A general theory of crime and deviance. Routledge, 2017.

[9] D. Garland and R. Sparks, “Criminology, social theory and the challenge of our times,” The British Journal of Criminology, vol. 40, no. 2, 2000, pp. 189-204.

[10] B. Hołyst, Kryminologia, ed. X, Warszawa, Wydawnictwo LexisNexis, 2009.

[11] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei and S. Savarese, “Social lstm: Human trajectory prediction in crowded spaces,” Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 961-971.

[12] L. Bontemps, J. McDermott and N. A. Le-Khac, “Collective anomaly detection based on long short-term memory recurrent neural networks,” in International Conference on Future Data and Security Engineering, Springer, Cham, 2016, pp. 141-152.

[13] Z. C. Lipton, J. Berkowitz and C. Elkan, „A critical review of recurrent neural networks for sequence learning,” arXiv preprint arXiv:1506.00019, 2015.

[14] H. Stoppiglia, G. Dreyfus. R. Dubois and Y. Oussar, “Ranking a Random Feature for Variable Feature Selection,” Journal of Machine Learning Research, vol. 3, 2003, pp. 1399-1414.

[15] Y. Zhang andY. Yang, “Cross-validation for selecting a model selection procedure,” Journal of Econometrics, vol. 187, 2015, pp. 95-112

[16] S. Chen; S. A. Billings andW. Luo, “Orthogonal least squares methods and their application to non-linear system identification,” International Journal of Control, vol. 50, no. 5, 1989, pp.1873 – 1896.

[17] T. Hastie, R. Tibshirani and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, 2008.

[18] I. Rivals and L. Personnaz, “On Cross-Validation for Model Selection,” Neural Computation, vol. 11, 1997, pp. 863-870.

[19] G. Monari and G. Dreyfus, “Withdrawing an example from the training set: an analytic estimation of its effect on a non-linear parameterized model,” Neurocomputing, vol. 35, 2000, pp. 195-201.

[20] G. Monari and G. Dreyfus, “Local Overfitting Control via Leverages,” Neural Computation, vol. 14, 2002, pp. 1481-1506.

[21] S. Jankowski, Z. Szymański, Z. Wawrzyniak, P. Cichosz, E. Szczechla and R. Pytlak, “Selection of neural networks for crime time series prediction by Virtual Leave One Out tests,” Proc. International Conference on Time Series and Forecasting (ITISE 2018), Granada, Sept. 2018, pp. 432-443.

[22] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrinka and J. Schmidhuber, “LSTM: A search space odyssey,”IEEE transactions on neural networks and learning systems, vol. 28, no. 10, 2017, pp. 2222-2232.

[23] Z. M.Wawrzyniak, C. Borowik, E. Szczechla, P. Michalak, R. Pytlak, P. Cichosz, D. Ircha, W. Olszewski and E. Perkowski, „Relationships Between Crime and Everyday Factors,” Proc. 2018 IEEE 22nd International Conference on Intelligent Engineering Systems (INES), June 2018, pp. 000039-000044.