Contextual Outlier Detection in Traffic Data

Inês Leite
Instituto Superior Técnico
Lisbon, Portugal

ABSTRACT

Despite the efforts placed by major European cities to optimize public transportation, traffic data analytics often disregard vital situational context. This work proposes a methodology to integrate situational context (including public events, planned interventions, and citizen notifications) in the analysis of public transport data. Major contributions are the: online consolidation and labeling of heterogeneous sources of context; calendar-driven statistical modeling of expected traffic behavior; forecasting; the integrative display of traffic and its situational context, accompanied by spatiotemporal navigation and zooming facilities, and outlier detection. Preliminary results collected from Lisbon's subway network system shows the relevance of these contributions to support context-sensitive decisions. This thesis makes a complete and thorough review of context-sensitive time series anomaly detection, an implementation of the algorithms and techniques used, and validation of the aforementioned methods and algorithms. Different approaches are explored to this end (modeling and forecasting traffic behavior), including neural networks and classic approaches based on auto-regressive moving averages and triple exponential smoothing.

KEYWORDS

contextual outlier detection, multivariate time series, forecasting, subway network traffic data

1 INTRODUCTION

Mobility in most European capital cities is not yet sustainable. In past years, the Lisbon City Council (CML) established acute efforts to collect all available traffic data and their situational context. The situational context consists of public events (including sports events, large-scale congresses, and cultural activities), interdictions (scheduled construction works, accidents, and citizen notifications), city urban planning maps, amongst other activities with potential impact on mobility. The integrative analysis of traffic data with these sources of situational context data offers unique opportunities for understanding traffic dynamics and, under the knowledge of upcoming events or scheduled activities, for both short and long-term public transport planning.

Despite the relevance of context-sensitive analysis of public traffic data, major drawbacks are typically observed. First, situational context is absent from traffic data analysis. Second, sources of situational context data are either private, dispersed or unavailable. Third, traffic data can be analyzed using multiple temporal-spatial-modal resolutions, leading to voluminous and hardly-usable results. Finally, ground-reference traffic behavior (to study meaningful traffic deviations) is not always adequately modeled. These challenges prevent a comprehensive, actionable and real-time understanding of traffic dynamics, leading to inefficiencies in the mobility system.

To address these challenges, this work laid on six major contributions:

(1) consolidation of traffic data from the Lisbon subway operator (METRO) with a comprehensive range of situational context data sources provided at the Open Data portal by the Lisbon City Council;
(2) online analytics, grounded on the updatable retrieval and automatic labeling (in accordance with the spatial, temporal and modal footprint) of context data;
(3) sound modeling of traffic behavior using calendar-guided statistical models for multivariate time series data;
(4) development of predictive algorithms for the analysis of public transport data in the presence of situational context information (with multivariate time series analysis and outlier detection).
(5) integrative display of (both expected and observed) traffic and its situational context, accompanied by temporal navigation and zooming facilities; and
(6) traffic data analytics with hierarchical views/facilities to support both citizen and operational decisions.

Owing to its importance, several time series forecasting methods have been proposed for the analysis of public traffic data in the presence of contextual outliers, such as the Holt-Winters, SARIMA and LSTM [45].

Context-sensitive analytics are critical to support public transport reinforcements, waiting time recommendations, sustainable transport planning, real-time messaging alerts, and data-centric coordination between authorities. The proposed contributions ensure extensibility to other traffic modalities and scalability to other cities in Europe, [20].

2 BACKGROUND

According to Brockwell in [7], a time series is a set of observations x_t, each one being recorded at a specific instant in time t.

If the time series has only one variable (n= 1) then this time series is referred to as univariate, if it has two variables or more (n >1) then it is referred to as multivariate [12]. The understanding of the structure and dependence of observations of a single (*univariate*) or several (*multivariate*) time series is achieved through *time series analysis* [5].

A multivariate time series $\mathbf{x}$ is a time series with n time points and m observations/variables per time point with $m > 1$:

$$\mathbf{x} = \begin{bmatrix} x_{11} & \dots & x_{1n} \\ x_{21} & \dots & x_{2n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \dots & x_{mn} \end{bmatrix}$$

A time series dataset is a set of time series x_{it}, where i is the variable and t is the time point. For simplicity sake, when considering univariate time series, each entry is represented by $x_{it} = x_t$.

A time series x_t, is assumed to be the sum of four components. *Time-trend*, which reflects a long-term movement of a time series. *Seasonal* component, which reflects a periodic movement in a series that repeats itself every year, month, etc. *Cyclical* component, which tracks the course of the business cycle, long term oscillations occurring in a time series. *Error*

or residuals component, which is the sum total of the effects of all those factors which are individually insignificant [21].

Classic Statistical Models

Differencing is a method of transforming a non-stationary time series into a stationary one. The first differencing value is the difference between the current time period and the previous time period. If these values fail to revolve around a constant mean and variance then we find the second differencing using the values of the first differencing. We repeat this until we get a stationary series.

By combining differencing with autoregression and a moving average model, we will obtain a non-seasonal **ARIMA** model. ARIMA is an acronym for Autoregressive Integrated Moving Average (in this context, integration is the reverse of differencing). The full model can be written as:

$$x'_t = c + \sum_{j=1}^{p} \alpha_j x'_{t-j} + \sum_{j=1}^{q} \beta_j \varepsilon_{t-j} + \varepsilon_t, \tag{1}$$

where x'_t is the differenced series [44]. The predictors on the right hand side include both lagged values of x_t and lagged errors. This is called an ARIMA(p,d,q) model, where:

p = order of the autoregressive part;
d = degree of first differencing involved;
q = order of the moving average part.

ARIMA models are also capable of modelling a wide range of seasonal data. A seasonal ARIMA model (**SARIMA**) is formed by including additional seasonal terms in the ARIMA models we have seen so far. It is written as follows [19]:

$$SARIMA \quad \underbrace{(p,\ d,\ q)}_{\substack{\uparrow \\ Non\ seasonal\ part \\ of\ the\ model}} \quad \underbrace{(P,\ D,\ Q)m}_{\substack{\uparrow \\ Seasonal\ part \\ of\ the\ model}}$$

where m = number of observations per unit of time. We use uppercase notation for the seasonal parts of the model, and lowercase notation for the non-seasonal parts of the model.

The seasonal part of the model consists of terms that are similar to the non-seasonal components of the model, but involve backshifts of the seasonal period. For example, a SARIMA $(1,1,1)(1,1,1)_{12}$ model learned for monthly data suggests a yearly seasonal cycle, and can be written as:

$$(1-\alpha_1 B)(1-\Phi_1 B^{12})(1-B)(1-B^{12})x_t = (1+\beta_1 B)\,(1+\Theta_1 B^{12})\varepsilon_t. \tag{2}$$

where B = backshifts of the seasonal period.

Although ARIMA methods are the most popular in time-series forecasting, exponential based methods such as the **Holt-Winters** models [23], have also shown to be useful prediction techniques.

Holt and Winters extended Holt's method to capture seasonality [18]. The **Holt-Winters** seasonal method (triple exponential smoothing) comprises the forecast equation and three smoothing equations — one for the level l_t, one for the trend b_t, and one for the seasonal component s_t, with corresponding smoothing parameters α, β^*, and γ. In this thesis, we use m to denote the frequency of the seasonality, i.e., the number of seasons in one month.

There are two variations to this method that differ in the nature of the seasonal component, such as the additive and multiplicative methods. The additive method is preferred when the seasonal variations are roughly constant through the series, while the multiplicative method is preferred when the seasonal variations are changing proportional to the level of the series.

The component form for the **additive** method is:

$$\hat{x}_{t+h} = \ell_t + hb_t + s_{t+h-m(k+1)} \tag{3}$$

$$\ell_t = \alpha(x_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \tag{4}$$

$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1} \tag{5}$$

$$s_t = \gamma(x_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}, \tag{6}$$

where k is the integer part of $(h-1)/m$, which ensures that the estimates of the seasonal indices used for forecasting come from the final year of the sample.

The level equation shows a weighted average between the seasonally adjusted observation $(x_t - s_{t-m})$and the non-seasonal forecast $(\ell_{t-1}+b_{t-1})$ for time t. The seasonal equation shows a weighted average between the current seasonal index, $(x_t - \ell_{t-1} - b_{t-1})$, and the seasonal index of the same season last year (i.e., m time periods ago).

The multiplicative model follows the same principles of the additive model, however the composition of the observed time series is given by the product of the components.

Machine Learning Approaches

In the context of the time series outlier detection, an outlier/or anomaly can be defined as a data point in a time series that is significantly different from the rest of the data points. Outliers are unusual observations that affect the analysis of the data and therefore must be treated with caution [5].

Outlier detection aims to find patterns in data that do not conform to expected behavior. Though contextual attributes are not directly related to the anomalous behavior, they provide useful information on contexts for outlier detection, [26].

In the context of this work, outlier detection approaches involve identifying the time of occurrence, which may not be known, as well as recognizing the type of outlier (local, global or contextual). In our work, we aim to detect outliers against available contextual information. Paradigmatic examples of contextual outlier include [39]:

- Time points whose anomalous behavior justified by the existence of a specific context;
- A point that presents abnormal behaviour compared to the population from which was drawn.

Deep Learning Approaches

A **neural network** can be thought of as a network of "neurons" which are organised in layers [19].

- Input layer: The activity of the input units represents the raw information that is fed into the network.
- Hidden layer(s): The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units.
- Output layer: The behaviour of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

The basic unit of computation in a neural network is the neuron. The **neuron** receives input from some other nodes and computes an output, [10]. Each input has an associated

weight (w), which is assigned on the basis of its relative importance to other inputs. The node applies an activation function f to the weighted sum of its inputs.

The function f is non-linear and is called the Activation Function. The purpose of the activation function is to introduce non-linearity into the output of a neuron. Every activation function (or non-linearity) takes a single number and performs a certain fixed mathematical operation on it.

RNNs are called recurrent because they perform the same task for every element of a sequence, with the output being depended on the previous computations, [24]. A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to a successor. The the formula for the current state, h_t, is given by:

$$h_t = f(h_{t-1}, x_t), \tag{7}$$

where h_t is the current timestamp, h_{t-1} is the previous time stamp, and x_t is the input state. Applying the activation function we have the equation, described as:

$$h_t = tanh(W_{hh}h_{t-1} + W_{hx}h_t), \tag{8}$$

where W is weight, h is the single hidden vector, W_{hh} is the weight at previous hidden state, W_{hx} is the weight at current input state, $tanh$ is the activation function. The output state is defined as:

$$h_t = W_{hy}h_t, \tag{9}$$

where Y_t is the output state, W_{hy} is the weight at the output state.

Sometimes, we only need to look at recent information to perform the present task. Recurrent Neural Networks suffer from short-term memory. If a sequence is long enough, they'll have a hard time carrying information from earlier time steps to later ones.

During back propagation, recurrent neural networks suffer from the **vanishing gradient** problem. Gradients are values used to update a neural networks weights. The vanishing gradient problem is when the gradient shrinks as it back propagates through time, [4].

Long Short Term Memory networks (**LSTM**) are a special kind of RNN, capable of learning long-term dependencies. LSTMs are explicitly designed to avoid the long-term dependency problem.

The first step in our LSTM is to decide what information we are going to throw away from the cell state. This decision is made by a sigmoid layer called the **forget gate**. It looks at the previous state h_{t-1} and the content input X_t and outputs a number between 0 and 1 for each number in the cell state C_{t-1}.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{10}$$

The next step is to decide what new information we're going to store in the cell state. First, the **input gate** decides which values we'll update. Next, a $tanh$ (activation function) layer creates a vector of new candidate values, $\hat{C}_t$. In the next step, we'll combine these two to create an update to the state.

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{11}$$

$$\hat{C}_t = tanh(W_c[h_{t-1}, x_t] + b_c) \tag{12}$$

We multiply the old state by f_t, forgetting the things we decided to forget earlier. Then we add $i_t * \hat{C}_t$. This is the new candidate values, scaled by how much we decided to update each state value.

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \tag{13}$$

The output will be based on our cell state, but will be a filtered version. We run a sigmoid layer which decides what parts of the cell state we are going to output. Then, we put the cell state through $tanh$ and multiply it by the output of the sigmoid gate, so that we only output the parts we decided to.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{14}$$

$$h_t = o_t * tanh(C_t) \tag{15}$$

3 STRUCTURED VIEW ON OUTLIER DETECTION

This proposes a taxonomic view on outlier detection in (multivariate) time series.

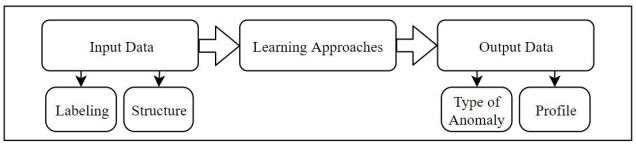

Figure 1: Structured view on outlier detection.

A key aspect of any outlier detection technique is the nature of the input data. **Input** is generally a collection of data instances (also referred as object, record, point, vector, pattern, event, case, sample, observation, or entity [9]). In this section we will focus on different input data types (Figure 2).

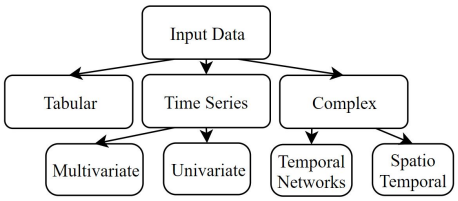

Figure 2: Different input data types.

A *tabular* data type, as the name implies is a data type that is structured in a tabular form. It arranges data elements in vertical columns and horizontal rows. [15].

Classic methods of outlier analysis assume the input data follows a tabular structure. In addition, a significant amount of work has been performed in time series outlier analysis. There are two different categories in time series outlier data analysis, which are:

- Univariate Data: Most of the earliest univariate methods for outlier detection rely on the assumption of an underlying known distribution of the data, which is assumed to be identically and independently distributed.
- Multivariate Data: In many cases, observations from a multivariate time series are incorrectly detected as outliers if each variable is considered independently. Outlier detection is possible only when multivariate analysis is performed, and the interactions among different variables are compared within the class of data.

Some studies find outliers considering only their temporal neighbors, while other work concerns outliers with respect to their spatial neighbors only. Combining these two notions, a *spatio temporal* outlier (ST-Outlier) is a spatio-temporal object whose behavioral/thematic attributes are significantly different from those of the other objects in its spatial and temporal neighborhoods [14].

The **labels** associated with a data instance denote whether that instance is *normal* or *anomalous* [9]. Labeling is often done manually by a human expert and so substantial effort is required to obtain the labeled training dataset. According to Hodge and Austin et al. [17] outlier detection techniques can operate in one of the following three modes:

- Supervised: Techniques trained in supervised mode assume the availability of a labelled training dataset. A typical approach in such cases is to build a predictive model for *normal* vs. *outlier* classes [9].
- Unsupervised: Techniques that operate in unsupervised mode do not require labelled training data [3].
- Semi-supervised: Techniques can be divided in three different categories: techniques that have some labeled instances for the normal class, or to the outlier class or both classes [37].

In accordance with the work by Chandola and Ben-Gal et al. [6, 9], we group **learning approaches** for outlier detection into:

Classification-based: According to Tang et al. [39], classification is used to learn a model (classifier) from a set of labeled data instances (training) and then, classify a test instance into one of the classes using the learnt model (testing). Classification-based outlier detection techniques operate in a similar two-phase method. The training phase learns a classifier using the available labeled training data. The testing phase classifies a test instance as normal or anomalous using the classifier.

Statistical: Statistical techniques fit a statistical model to the given data and then apply a statistical inference test to determine if an unseen instance belongs to the model or not, [31]. Instances that have a low probability to be generated from the learned model, based on the applied test statistic, are declared as outliers. Both parametric, as well as non-parametric techniques, have been applied to fit a statistical model.

Clustering: is used to group similar data instances into clusters. Clustering is primarily an unsupervised technique though semi-supervised clustering approaches for outlier detection has also been explored lately, [5]. Even though clustering and anomaly detection appear to be fundamentally different from each other, several clustering-based anomaly detection techniques have been developed. Cluster structure can be easily destroyed by few outliers; on the contrary, the outliers are defined by the concept of a cluster, which is recognized as the points belonging to none of the clusters or small clusters.

Spectral: Spectral techniques try to find an approximation of the data using a combination of attributes that capture the bulk of variability in the data [33]. Thus the general approach adopted by spectral anomaly detection techniques is to determine such subspaces in which the anomalous instances can be easily identified. Such techniques can work in an unsupervised as well as semi-supervised setting.

The **output** of an outlier detection algorithm can be described based on two categories:

Type of Outlier: An important aspect of an outlier detection technique is the nature of the desired outlier. Outliers can be classified into following three categories [9]:

- Local Outlier: If an individual data instance can be considered as anomalous with respect to the rest of data, then the instance is termed as a local outlier. This is the simplest type of outlier and is the focus of majority of research on outlier detection [37].
- Global Outlier: If a collection of related data instances is anomalous with respect to the entire dataset, it is termed as a collective outlier. The individual data instances in a collective outlier may not be outliers by themselves, but their occurrence together as a collection is anomalous [37].
- Contextual Outlier: If a data instance is anomalous in a specific context, but not otherwise, then it is termed a contextual outlier [9]. The notion of a context is induced by the structure in the dataset and has to be specified as a part of the problem formulation. Each data instance is defined using the following two sets of attributes: contextual attributes and behavioral attributes (non contextual characteristics of an instance).

Profile: Another aspect for any outlier detection technique is the manner in which the outliers are reported. Typically, the outputs produced by outlier detection techniques are one of the following two types [9]:

- Numerical: Scoring techniques assign an anomaly score to each instance in the test data depending on the degree to which that instance is considered an outlier. Thus the output of such techniques is a ranked list of outliers. An analyst may choose to either analyze the top few outliers or use a cutoff threshold to select the outliers.
- Categorical: Techniques in this category assign a label (normal or anomalous) to each test instance. Scoring-based outlier detection techniques allow the analyst to use a domain specific threshold to select the most relevant outliers [37].

4 RELATED WORK

In this section, relevant work regarding time series prediction, outlier detection and contextual outlier detection are brought up and discussed.

Time Series Prediction

Time Series Prediction is a function approximation task whose goal is to estimate future values of a sequence of observations based on current and past values of the sequence, [19]. In addition to traffic prediction, forecasting has been applied for reliability monitoring, predictive maintenance and intrusion detection, and can effectively improve availability, security, and the overall service experience [13].

In accordance with Et al. [28], there are two basic types of time series forecasting: univariate (simple regression) and multivariate (multivariate regression). Autoregressive (AR), moving average (MA), and autoregressive moving average (ARMA) models are often used for the purpose of time-series modeling, analysis and prediction. These models have been successfully used in a wide range of applications such as speech analysis, noise cancellation, stock market analysis, among others [7, 16, 27, 46].

Kalekar et al. [23] and Hyndman et al. [19], concentrate on the analysis of seasonal time series data using Holt-Winters exponential smoothing methods.

Wong et al. [42], proposed a novel adaptive neural network (ADNN) with the adaptive metrics of inputs and a new mechanism for admixture of outputs for time-series prediction. The proposed ADNN method can predict periodical time-series with a complex structure. Guo in [13] propose an adaptive gradient learning method for recurrent neural networks (RNN) in the context of online learning problem.

Outlier Detection

Patcha and Park in [34] provide a comprehensive survey of anomaly detection systems and hybrid intrusion detection systems. Aggarwal et al. [1] notes that outliers may be considered as noise points lying outside a set of defined clusters or alternatively outliers may be defined as the points that lie outside of the set of clusters but are also separated from the noise.

Singh et al. [37] distinguish simple outliers from complex outliers and define two types of complex outliers, contextual and collective outliers. In [2], Agyemang discuses practical applications of outlier mining, and provides a taxonomy for categorizing related mining techniques.

According to [9] outlier detection refers to the problem of finding patterns in data that do not conform to expected normal behavior. These anomalous patterns are often referred to as outliers, anomalies, discordant observations, faults, noise, errors, damage, novelty, peculiarities, etc. in different application domains.

Hodge and Austin et al. [17] provide an extensive survey of anomaly detection techniques developed in machine learning and statistical domains. Jordaan et al. [22] explore the use of a robust model-based outlier detection approach that makes use of the characteristics of the support vectors obtained by the Support Vector Machine method.

Although the application domain of some of the mentioned contributions is different from ours, some of the principles can be mapped for urban traffic analysis.

Outlier Detection using Neural Networks

Deep neural networks have been studied extensively for solving supervised learning tasks such as classification and prediction. However, only few studies consider deep neural networks for resolving unsupervised learning tasks such as outlier detection.

In [36] Shipmon and Gurevitch trained various models including DNNs, RNNs, and LSTMs to perform regression and predict the expected value in the time series. Secondly they created anomaly detection rules that compared the actual values to predicted values. Xia in [43] studies removing outliers from noisy image data. In particular, a deep auto-encoder separates inliers from outliers according to reconstruction errors. This study shows that deep neural networks are capable of being more accurate than traditional methods such as LOF and One-class SVM.

Lu in [29] proposes a deep neural network to detect outliers in video, which outperforms existing, non-deep learning based methods. They propose an auto-encoder that comprises two components—a traditional feed forward neural network to extract local features, and an LSTM neural network to compress sequential features.

Context Incorporation

There are several factors derived from context that can influence the mobility behavior. Some of these factors are planned and others are not. The latter cases encompass weather, air pollution, humidity, disease spread, etc, mentioned in [38], and the former cases encompass football matches, concerts, festivals, etc ([40], [25]).

Planned special events, term introduced by Latoski et al. [11], are even more challenging to deal with due to the dependence of other factors (infrastructure, audience, among others).

The work present by Soua et al. [38] proposes a framework to address the traffic flow prediction. This work also integrate several data sources, including event-based data from Twitter and weather. Rodrigues et al. [35], propose a Bayesian additive model with Gaussian process components to predict the number of public transport's trips to special event areas.

Kwoczek et al. [25] focus on the prediction of traffic congestion in the presence of planned special events, based on K-NN algorithm and on the mapping of likely congestion road segments due to planned special events.

Contextual Outlier Detection

Tang et al. [39], define a contextual outlier like a small group of objects that share strong similarity with a significantly larger reference group of objects on some attributes, but deviate dramatically on some other attributes. This paper focus on improving the interpretability of outliers.

Wang's approach in [41] has several advantages including producing outlier scores which can be interpreted as stationary expectations and their calculation in closed form in polynomial time. Also, Wang shows that point outlier detection using stationary distribution is a special case of contextual outlier detection. This approach allows to find both global and contextual outliers simultaneously and create a meaningful ranked list consisting of both types of outliers.

5 SOLUTION

The traffic modality considered to motivate our contributions is the Lisbon's subway network of transportation (validation at the entry and exit stations) provided by METRO under an agreement between INESC-ID and the Lisbon City Council (CML) in the context of the ILU project. The process can be briefly described in the Figure 3.

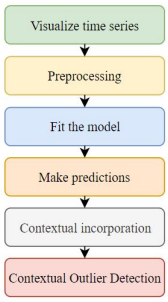

Figure 3: Proposed methodology.

The subway network currently has 50 stations distributed in four lines, operating from 6am to 1am. Illustrating, when considering cumulative validations every 15 minutes, subway traffic data (passenger volume data) is seen as multivariate time series data with 75 data points per day (6am–1am) and multivariate order dependent on stations or station-groups of interest.

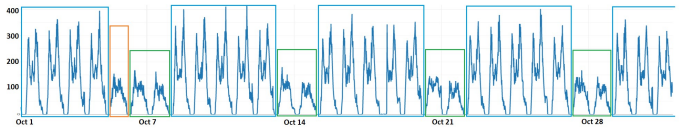

Figure 4: Passenger volume at the Alameda station (cumulative entry validations every 15 minutes) during October 2018.

Figure 4 depicts the total amount of entrances at Alameda station from during October, 2018. It is noticeable calendar-specific aspects, such as different usage patterns between work days (blue) and weekends (green), and also a holyday in the Friday of the first week (orange). The analysis of the series was conducted using Python.

To preprocess the data, the missing values were imputed. There are many options we could consider when replacing a missing value, we decided to replace missing values using interpolation weighted by calendar-driven auto-regression. On October 18, the Lisbon underground was on strike from 06:30 to 9:30 a.m. These data points were replaced using the interpolation described, as they would decrease model correctness since there is no passenger volume. Deviations from the daily average were maintained to correctly identify the outliers in the dataset. To maintain the timeliness of the data, the Lisbon subway closing period was removed.

Observed versus modeled traffic behavior was assessed using:

- Different methods (including Holt-Winders, SARIMA and LSTM)
- Small vs. lengthy time series (from 72 to 2232 points, parameter m)
- Calendar free vs. specific (e.g. day of week, work day, holiday, undifferentiated)
- Diverse temporal granularities (windows of minutes to hours) and spatial granularities (station-specific, region-specific, line-specific and all-network)

The *Holt-Winters* model, called also the triple exponential smoothing model, is a well-known adaptive model used to modeling time series characterized by trend and seasonality. Since the data have a constant seasonality (for example, a daily set of m= 75 periods of 15 minutes each), we opted by the additive model. The additive model is the best Holt-Winters model when seasonality is constant and does not change proportionally.

Before started building the model, first it is required to know how to estimate model parameters automatically. For that, we have to choose a loss function suitable for the task. Then using cross-validation we evaluate our chosen loss function for given model parameters, calculate the gradient, adjust model parameters and so forth, bravely descending to the global minimum of error.

The question is how to do cross-validation on time series, because time series do have time structure and one just can't randomly mix values in a fold without preserving this structure, otherwise all-time dependencies between observations will be lost. Given this context, we used a less-trivial approach to optimize the model parameters called cross-validation on a rolling basis.

We train our model on a small segment of the time series, from the beginning until some t, make predictions for the next $t + n$ steps and calculate an error. Then we expand our training sample until $t + n$ value and make predictions from $t + n$ until $t + 2n$, and we continue moving our test segment of the time series until we hit the last available observation. As a result, we have as many folds as many n will fit between the initial training sample and the last observation.

Knowing how to set cross-validation, we found optimal parameters for the Holt-Winters model, recall that we have daily seasonality, hence the m = 75 parameter. In the Holt-Winters model, there's a constraint on how big smoothing parameters could be, each of them is in the range from 0 to 1, therefore to minimize loss function we had to choose an algorithm that supports constraints on model parameters (Truncated Newton conjugate gradient et al. [32]).

The appropriate Holt-Winters smoothing factors (alpha= α, beta= β and gamma= γ) for best fit model was chosen based on least Mean Squared Error values.

Smoothing factors			
Smoothing parameters	α	β	γ
Values	0.0616	0.0678	0.02338

Table 1: Holt-Winters smoothing parameters.

The presence of seasonality in the series is evident from gamma value. This value justifies once again the choice of the additive model. Holt-Winter's Exponential Smoothing accurate on forecasting seasonal time series data, either it's pattern shows trend or not.

After describing the data, it is clear that we have a time series with a seasonal component, so it makes sense to use a *SARIMA* model. To do this we need to choose p, d, q values for the ARIMA, and P, D, Q values for the Seasonal component [7]. There are many ways to determine these values statistically, such as looking at auto-correlation plots, correlation plots, etc.

One simple approach is to perform a grid search over multiple values of p, d, q, P, D, and Q using some sort of performance criteria. The Akaike Information Criterion (AIC) is an estimator of the relative quality of statistical models for a given set of data. Given a collection of models for the data, AIC estimates the quality of each model, relative to each of the other models. The AIC value will allow us to compare how well a model fits the data and takes into account the complexity of a model, so models that have a better fit while using fewer features will receive a better (lower) AIC score than similar models that utilize more features.

Auto arima function allows us to set a range of p, d, q, P, D, and Q values and then efficiently fit models for the most promising combinations. Then the model will keep the combination that reported back the best AIC value.

The idea associated with the *LSTM* is that each computational unit is linked not only to a hidden state h but also to a state c of the cell that plays the role of memory. The change from c_{t-1} to c_t is done by constant gain transfer equal to 1, so that errors are propagated at previous steps without any gradient disappearance phenomenon.

As previously mentioned in our Solution, we forecast based on three different models: Holt-Winters, SARIMA and LSTM. To predict we need a start and end date or index to be specified.

A confidence interval is a range of values based on a point estimate that contains the true population parameters at some confidence level. Confidence intervals measure deviation for each time point in the seasonal cycle and this mechanism bases on expected seasonal variability. Holt-Winters and SARIMA methods were first used to model and forecast.

Confidence interval is computed by comparing last period of collected subway traffic values with fitted model values for

the same period. Subtract of real and predicted values is next scaled with the factor estimated by Holt-Winters function. The obtained value is finally multiplied by scaling factor (φ).

Brutlag method was included into the model to build confidence intervals:

$$\hat{x}_{max_t} = l_{t-1} + b_{t-1} + s_{t-T} + \varphi * d_{x-T} \quad (16)$$

$$\hat{x}_{min_t} = l_{t-1} + b_{t-1} + s_{t-T} - \varphi * d_{x-T} \quad (17)$$

$$d_x = \gamma|x_t - \hat{x}_t| + (1-\gamma) * d_{x-T}, \quad (18)$$

where T is the length of the season, d is the predicted deviation, x_{max} is the upper bound, x_{min} is the lower bound of Brutlag's confidence band, and the other parameters were taken from the triple exponential smoothing.

Confidence interval width is controlled with scale parameter is computed with scale parameter value of 2.96 (φ). Brutlag proposes et al. [8] sensible values of scale parameter are between 2 and 3.

The use of confidence intervals is interesting to check the deviations and make an analysis of the time series boundaries. In this thesis we use confidence intervals only for classical methods.

A simple mechanism to detect an anomaly is to check if an observed value of the time series falls outside the confidence band. However, this mechanism often yields a high number of false positives. A more robust mechanism is to use a moving window of a fixed number of observations.

If the number of violations (observations that fall outside the confidence band) exceeds a specified threshold, then trigger an alert for aberrant behavior. Formally, define a violation as an observation x_t that falls outside the interval:

$$(l_{t-1} + b_{t-1} + s_{t-T} - \varphi * d_{x-T}, l_{t-1} + b_{t-1} + s_{t-T} + \varphi * d_{x-T}) \quad (19)$$

Define a failure as exceeding a specified number of threshold violations within a window of a specified numbers of observations (the window length - φ).

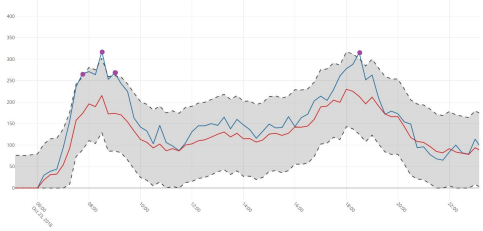

Figure 5: Outlier Detection using Confidence Intervals.

With a prediction length of l, each of the selected d dimensions of $x_t \in X$ for $l < t \leq n$ - l is predicted l times. We compute an error vector $e(t)$ for point x_t as $e(t)$, where $e(t)$ is the difference between x_t and its value as predicted.

$$e = x_t - \hat{x}_t \quad (20)$$

where x_t and $\hat{x}_t$ are respectively the observed (true) and predicted values. The prediction model trained on sN is used to compute the error vectors for each point in the validation and test sequences. The error vectors are modelled to fit a Gaussian distribution to error vectors computed over test data by the maximum likelihood estimation $N = N(\mu, \sum)$.

The likelihood $p(t)$ of observing an error vector $e(t)$ is given by the value of N at $e(t)$. The error vectors for the points from $vN1$ are used to estimate the parameters μ and $\sum$ using Maximum Likelihood Estimation. An observation x_t is classified as "anomalous" if $p(t) < T$, else the observation is classified as normal. The sets $vN2$ and vA are used to learn T by maximizing F_β-score (where anomalous points belong to positive class and normal points belong to negative class) [30].

We can measure the rarity of the event with the location in the distribution. The Mahalanobis' distance is here seen as a statistic representing an anomaly score. Assuming the parameters of a M dimensional Gaussian distribution are estimated as follows:

$$p(x|Data) = N(x|\hat{\mu}, \sum) \quad (21)$$

Then, the Mahalanobis' distance is defined as:

$$a(x) = (x - \hat{\mu})^T \hat{\sum} (x - \hat{\mu}) \quad (22)$$

We can measure the rarity of the event with a(x).

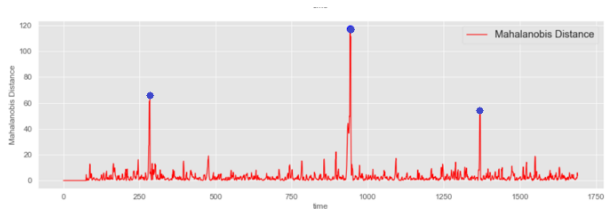

Figure 6: Outlier Detection using Mahalanobis' distance.

The proposed Solution builds upon the already established efforts of the Lisbon City Council (CML) to collect all relevant events taking place in the city. These events are stored using semi-structured representations (JSON) at the *Lisboa Aberta* portal [1]. In this work, we consider: congresses [2], sport events, Lisbon cultural agenda (concerts, exhibitions and other activities [3]), accidents, scheduled construction works, and citizen notifications at *Na minha rua* portal.

A periodic routine to read new information from these diverse and heterogeneous sources of situational context data is implemented for an automatic and integrative display of traffic data and their situational context. In addition, the retrieved events are automatically annotated in accordance with their category and the duration of the event is placed in accordance with the information available or augmented in the presence of user rules. One illustrative rule is the specification that sport events approximately impacts entry validations at public transport 60min-5min before the game and 0-30 mins after the game.

When a single data point (e.g., value from a time series) is declared as an outlier because of its relationship to its related data items, it is referred to as a contextual outlier or anomaly. This is because such an outlier can only be understood in the context of its relationship with items in the temporal neighborhood.

[1] http://dados.cm-lisboa.pt/tr/dataset
[2] http://dados.cm-lisboa.pt/tr/dataset?q=espaco+publico
[3] http://dados.cm-lisboa.pt/tr/dataset?q=agenda+cultural

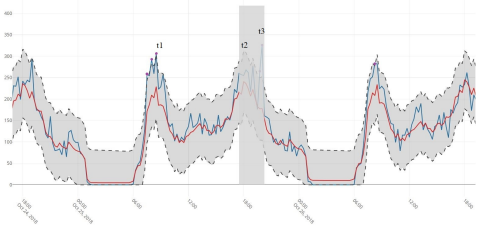

Figure 7: Example of context outlier detection in our Solution.

Given the contextual outlier analysis, we can get three different types of observations to take into account in our Solution (Figure 7):

- Outliers: A data point in a time series, i.e. $t1$, that the observed value falls outside the confidence band. This point is significantly different from the rest of the data points.
- Contextual Points: A data point in a time series, i.e. $t2$, that the observed value falls inside the context-specific band.
- Contextual Outliers: A data point in a time series, i.e. $t3$, that the observed value falls inside the context-specific band and outside the confidence band. In this context, we assume that confidence bands relax due to the contextual-specific situation in which these points find themselves.

6 RESULTS

In this chapter, we undertake a comprehensive analysis to identify the best statistical models of subway usage from multivariate time series data. Mean absolute error (MAE), root mean squared error (RMSE) and symmetric mean absolute percentage error (SMAPE) were collected to validate the different models.

The results of the contextual outlier detection algorithm are presented. Outliers were obtained by modeling the algorithms described in Section 2 (SARIMA, Holt-Winters and LSTM), and the respective confidence intervals. These outliers were then interspersed with context-specific data to separate false from true contextual outliers.

	SARIMA			Holt-Winters			LSTM		
	RSME	SMAPE	MAE	RSME	SMAPE	MAE	RSME	SMAPE	MAE
Alameda	30.18	23.02	22.07	19.48	49.84	13.03	26.26	18.29	18.69
Aeroporto	24.89	26.49	17.25	20.06	50.25	13.23	22.77	26.65	16.85
Campo Grande	50.31	25.07	35.24	15.33	49.66	10.24	44.39	22.61	31.44
Red Line	156.63	13.74	120.42	221.94	47.95	103.47	182.65	15.34	119.15
Subway Network	746.30	13.37	535.98	1675.51	69.87	1041.28	930.43	14.64	556.64

Table 2: Forecasting errors for SARIMA, Holt-Winters and LSTM algorithms.

The model's forecast accuracies were calculated, tested and compared using MSE, MAE, and SMAPE. This thesis has compared the forecasting ability of Holt-Winters, SARIMA and LSTM models with respect to traffic data. The study results demonstrate that both models are effective, however Holt-Winters model seems to be a more precise and accurate model.

Table 2 represents the values obtained with the three different models analysed. The forecast horizon used was the working days of the month of October 2018.

From table 2 we understand that Holt-Winters model has the minimum RMSE and SMAPE values when compared with SARIMA and LSTM models. The LSTM model has the minimum MAE value. The Holt-Winters model's relative ease of use makes the model useful in forecasting comprehensive trends. The figure shows the errors collected for each algorithm in Alameda station.

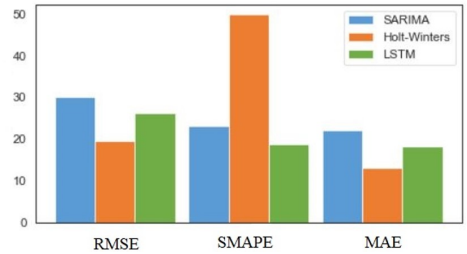

Figure 8: Algorithms bar chart comparison.

We can conclude that:

- Classical time series forecasting methods (SARIMA and Holt-Winters) focus on univariate data with linear relationships and fixed and manually-diagnosed temporal dependence.
- The number of training times, known as "epoch" in deep learning, has effect on the performance of the trained forecast model.
- Classical methods like SARIMA focus on fixed temporal dependence: the relationship between observations at different times, which necessitates analysis and specification of the number of lag observations provided as input.
- Machine learning and deep learning methods do not yet deliver on their promise for univariate time series forecasting.
- As LSTMs are equipped to to learn long term correlations in a sequence, they can model complex multivariate sequences without the need to specify any time window.

To analyze the results of contextual outlier detection, we selected 3 context-specific events that took place in the city of Lisbon, which are: Benfica game, 7 October 2018 (Entry validation hours: 3 p.m. to 5:15 p.m., Exit validation hours: 7 p.m. to 8:30 p.m.), Sporting game, 25 October 2018 (Entry validation hours: 3:45 p.m. to 5:45 p.m , Exit validation hours: 7 p.m. to 8:15 p.m.) and Sporting game, 28 October 2018 (Entry hours: 6 p.m to 7:45 p.m. , Exit validation hours: 9:30 p.m. to 10:45 p.m.)

The expected/reference traffic behavior appears to be robustly estimated given the high variability of the time series and the fact that we did not exclude context-specific variations from the estimations.

In periods without situational context, RMSE and MAE generally account for the less than 7% of the real/observed traffic. We hypothesize the error to nearly fully explained by daily/irregular variations and context-specific variations.

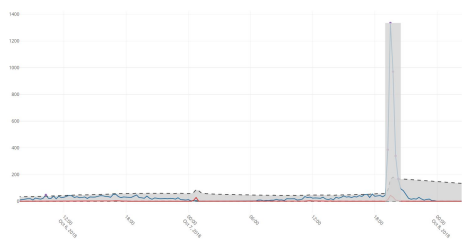

Figure 9: Contextual outlier detection in Alto dos Moinhos station.

Figure 9 shows the impact that sport events (event at the Benfica stadiums) have in the subway traffic of specific station.

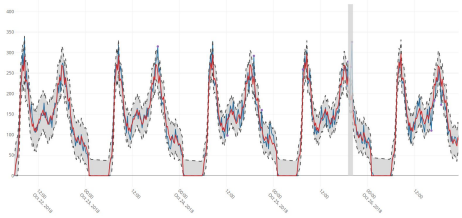

Figure 10: Contextual outlier detection in Alameda station.

Although the Alameda station is not directly correlated with any of the sports events, by modeling it found the presence of contextual outliers in the entry validation data (Figure 10). The same can be justified as the passenger volume of fans moving to the stadium before the sports event starts.

To analyze the performance of our solution, we compared the number of outliers with the number of time points with a specific context. For this, we selected intervals in which the described sportive events occurred, thus obtaining the following table:

	Alto dos Moinhos	Campo Grande	Alameda
Type of Validation	Exit validation	Entry validation	Entry validation
# Data Points	150	150	375
# Outliers	6	7	11
# Contextual points	5	8	4
# Contextual Outliers	5	7	3

Table 3: Contextual outlier detection results.

Table 3 represents the results obtained at Alto dos Moinhos, Campo Grande and Alameda stations, with a Brutlag factor of 2.96.

Both algorithms to contextual outlier detection identify the main existing outliers due to sporting events. We can conclude that the algorithm that uses confidence intervals is more accurate since it is possible to obtain results with 95% confidence. The second algorithm just detects outliers more deviant from the standard and is not that accurate.

The following list presents a few possible areas of improvements to further improve our Solution and fulfil the aims of the project.

- **Selection of Data:** Only three out of the fifty provided time series with different characteristics were selected as representatives for evaluating the outlier detection algorithm.
- **Context data:** A limitation of our Solution is the lack of any labeled contextual data and a possible next step would be testing the algorithm on labeled data.
- **Improvement of the algorithm:** From the current state of the algorithm, the greatest improvement is probably to refine the features that distinguish normal instances from abnormal ones.
- **Design of the visualization tool:** The outlier detection algorithm was implemented with the intention to be user friendly [20]. An important part for the final application to be user friendly is the design of the user interface.
- **Multivariate Analysis:** One possible improvement could be looking at the stations as a multivariate time series to detect possible dependencies and correlations.

7 CONCLUSIONS

The existing research on outlier detection for time series is limited to developing techniques that are suitable for specific application domains. In this thesis we provided a comprehensive understanding and structured overview of the research on anomaly detection techniques for time series. Our study reveals that the performance of the anomaly detection techniques is closely tied to the nature of the underlying data, and hence the techniques exhibit varying performance across application domains.

The most important and strongest statement we can make after concluding this work is a novel approach, using a traditional approach and a machine learning based approach. Under these two fields, given that the data is time-indexed, time series analysis was conducted, with the objective of detect outliers. Given this, this tool is completely automated, i.e., does not require human intervention in the entire process and can make predictions into the future.

To conclude, the implications of this work are various. Firstly, a complete, end-to-end dashboard, capable of giving a complete snapshot of the present state of how the METRO flow of passengers is occurring, secondly, a complete review of anomaly detection methods and algorithms, and lastly but not least, an implementation of the chosen algorithms, SARIMA and the Holt-Winters methods, and overall contextual outlier detection methodology.

Four major future directions are identified:

- the extension of the proposed methodology for multimodal traffic data analysis;
- the context-sensitive modeling of expected behavior based on the selective recollection of historical data with analogous context;
- modeling station-wise dependencies to improve the learning process;
- the context-sensitive forecasting of traffic data in the presence of planned events.

REFERENCES

[1] Charu C. Aggarwal. 2016. Time Series and Multidimensional Streaming Outlier Detection. In *Outlier Analysis*. Springer International Publishing, Cham, 273–310. https://doi.org/10.1007/978-3-319-47578-3{_}9

[2] Malik Agyemang, Ken Barker, and Rada Alhajj. 2006. A comprehensive survey of numeric and symbolic outlier mining techniques. *Intelligent Data Analysis* 10, 6 (11 2006), 521–538. https://doi.org/10.3233/IDA-2006-10604

[3] Samaneh Aminikhanghahi and Diane J. Cook. 2017. A survey of methods for time series change point detection. *Knowledge and Information Systems* 51, 2 (5 2017), 339–367. https://doi.org/10.1007/s10115-016-0987-z

[4] AG Barto and MI Jordan. 1992. Gradient following without backpropagation in layered networks. *et-al. Frontiers in cognitive neuroscience* (1992), 443–449.

[5] Sabyasachi Basu and Martin Meckesheimer. 2007. Knowledge and Information Systems Automatic outlier detection for time series: an application to sensor data. *Knowl Inf Syst* 11, 2 (2007), 137–154. https://doi.org/10.1007/s10115-006-0026-6

[6] Irad Ben-Gal. 2005. Outlier Detection. In *Data Mining and Knowledge Discovery Handbook*, Maimon Oded, , and Lior Rokach (Eds.). Springer US, Boston, MA, 131–146. https://doi.org/10.1007/0-387-25465-X_7

[7] Peter J. Brockwell and Richard A. Davis. 2016. *Introduction to Time Series and Forecasting*. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-29854-2

[8] Jake D. Brutlag. 2000. Aberrant Behavior Detection in Time Series for Network Monitoring. , 139–146 pages.

[9] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection. *Comput. Surveys* 41, 3 (7 2009), 1–58. https://doi.org/10.1145/1541880.1541882

[10] Rodrigo De Oliveira, Ramon Cristian Fernandes Araújo, Fabrício Barros, Adriano Segundo, Ronaldo Zampolo, Wellington Fonseca, Victor Dmitriev, and Fernando Brasil. 2017. A System Based on Artificial Neural Networks for Automatic Classification of Hydro-generator

Stator Windings Partial Discharges. *Journal of Microwaves, Optoelectronics and Electromagnetic Applications* 16 (09 2017), 628–645. https://doi.org/10.1590/2179-10742017v16i3854

[11] WM Dunn and SP Latoski. 2003. Managing travel for planned special events. In *Institute of Transportation Engineers (ITE) 2003 Technical Conference and ExhibitInstitute of Transportation Engineers (ITE)*.

[12] Bilal Esmael, Gerhard Thonhauser, Arghad Arnaout, and Rudolf K Fruhwirth. 2012. Multivariate Time Series Classification by Combining Trend-Based and Value-Based Approximations Mahmamt-Monitoring and Analysis of Heavy Machinery using non-intrusive measurement techniques View project Optimization of Core Tripping View project Multivariate Time Series Classification by Combining Trend-Based and Value-Based Approximations. *LNCS* 7336 (2012). https://doi.org/10.1007/978-3-642-31128-4{_}29

[13] Tian Guo, Zhao Xu, Xin Yao, Haifeng Chen, Karl Aberer, and Koichi Funaya. 2016. Robust Online Time Series Prediction with Recurrent Neural Networks. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 816–825. https://doi.org/10.1109/DSAA.2016.92

[14] Manish Gupta, Jing Gao, Charu C. Aggarwal, and Jiawei Han. 2014. Outlier Detection for Temporal Data: A Survey. *IEEE Transactions on Knowledge and Data Engineering* 26, 9 (9 2014), 2250–2267. https://doi.org/10.1109/TKDE.2013.184

[15] Marc Gyssens and Laks V S Lakshmanan. 1997. A Foundation for Multi-Dimensional Databases. In *A Foundation for Multi-dimensional Databases (VLDB '97)*. 23rd International Conference on Very Large Data Bases, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 106–115.

[16] James D. (James Douglas) Hamilton. 1994. *Time series analysis*. Princeton University Press, Princeton N.J. 799 pages. https://www.worldcat.org/title/time-series-analysis/oclc/28257560

[17] Victoria J Hodge and Jim Austin. 2004. *A Survey of Outlier Detection Methodologies*. Technical Report. Department of Computer Science, University of York, York.

[18] Charles Holt. 1957. *Forecasting seasonals and trends by exponentially weighted moving averages*. Carnegie Institute of Technology Graduate school of Industrial Administration, Pittsburgh Pa.

[19] Robin John Hyndman and George Athanasopoulos. 2018. Forecasting: Principles and Practice.

[20] Rui Henriques Inês Leite, Anna Carolina Finamore. 2019. Context-sensitive analysis of public transport data. *Proceedings of 8th Transport Research Arena TRA 2020* (2019).

[21] Raj K. Jain. 2001. *Monthly Labor Review Seasonal Adjustment A state space model-based method of seasonal adjustment*. Technical Report. 124 Monthly Lab. Rev. 37 (2001). https://www.bls.gov/opub/mlr/2001/07/art4full.pdf

[22] Elsa M. Jordaan and Guido E Smits. 2004. Robust outlier detection using SVM regression. In *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat No 04CH37541) IJCNN-04*. IEEE, 2017–2022. https://doi.org/10.1109/IJCNN.2004.1380925

[23] Prajakta S Kalekar. 2004. *Time series Forecasting using Holt-Winters Exponential Smoothing*. Technical Report. Kanwal Rekhi School of Information Technology. https://pdfs.semanticscholar.org/1eae/285bd8d62eaf9dc37e8d2351845ba2903664.pdf

[24] Andrej Karpathy, Justin Johnson, and Fei-Fei Li. 2015. Visualizing and Understanding Recurrent Networks. *CoRR* abs/1506.02078 (2015). arXiv:1506.02078 http://arxiv.org/abs/1506.02078

[25] Simon Kwoczek, Sergio Di Martino, and Wolfgang Nejdl. 2014. Predicting and visualizing traffic congestion in the presence of planned special events. *Journal of Visual Languages & Computing* 25, 6 (2014), 973 – 980. https://doi.org/10.1016/j.jvlc.2014.10.028 Distributed Multimedia Systems DMS2014 Part I.

[26] Jiongqian Liang and Srinivasan Parthasarathy. 2016. Robust Contextual Outlier Detection: Where Context Meets Sparsity. *CoRR* abs/1607.08329 (2016). arXiv:1607.08329 http://arxiv.org/abs/1607.08329

[27] Chenghao LIU, Steven C. H. HOI, Peilin ZHAO, and Jianling SUN. 2016. Online ARIMA algorithms for time series prediction. *AAAI-16: Proceedings of Thirtieth AAAI Conference on Artificial Intelligence: Phoenix, Arizona USA, 2016 February 12-17* (1 2016). https://ink.library.smu.edu.sg/sis_research/3619

[28] D. Lowe and A.R. Webb. 1991. Time series prediction by adaptive networks: a dynamical systems perspective. *IEE Proceedings F Radar and Signal Processing* 138, 1 (1991), 17. https://doi.org/10.1049/ip-f-2.1991.0004

[29] W. Lu, Y. Cheng, C. Xiao, S. Chang, S. Huang, B. Liang, and T. Huang. 2017. Unsupervised Sequential Outlier Detection With Deep Architectures. *IEEE Transactions on Image Processing* 26, 9 (Sep. 2017), 4321–4330. https://doi.org/10.1109/TIP.2017.2713048

[30] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. 2015. Long Short Term Memory Networks for Anomaly Detection in Time Series. In *ESANN*.

[31] Markos Markou and Sameer Singh. 2003. Novelty detection: a review—part 1: statistical approaches. *Signal Processing* 83, 12 (2003), 2481 – 2497. https://doi.org/10.1016/j.sigpro.2003.07.018

[32] Stephen G. Nash. 2000. A survey of truncated-Newton methods. *J. Comput. Appl. Math.* 124, 1 (2000), 45 – 59. https://doi.org/10.1016/S0377-0427(00)00426-X Numerical Analysis 2000. Vol. IV: Optimization and Nonlinear Equations.

[33] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. 2001. On Spectral Clustering: Analysis and an Algorithm. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic (NIPS'01)*. MIT Press, Cambridge, MA, USA, 849–856.

[34] Animesh Patcha and Jung-Min Park. 2007. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks* 51, 12 (8 2007), 3448–3470. https://doi.org/10.1016/j.comnet.2007.02.001

[35] Filipe Rodrigues, Stanislav Borysov, Bernardete Ribeiro, and Francisco Pereira. 2016. A Bayesian Additive Model for Understanding Public Transport Usage in Special Events. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP (12 2016), 1–1. https://doi.org/10.1109/TPAMI.2016.2635136

[36] Dominique T Shipmon, Jason M Gurevitch, Paolo M Piselli, and Steve Edwards. 2017. *Time Series Anomaly Detection Detection of Anomalous Drops with Limited Features and Sparse Examples in Noisy Highly Periodic Data*. Technical Report. Google, Inc. https://static.googleusercontent.com/media/research.google.com/pt-PT//pubs/archive/dfd834facc9460163438b94d53b36f51bb5ea952.pdf?fbclid=IwAR2HalwKs7jSY5mPJerHOvdMrzYlxj99tHk8PArfEKYf2tugr73So-SMviI

[37] Karanjit Singh and Shuchita Upadhyaya. 2012. *Outlier Detection: Applications And Techniques*. Technical Report. Department of Computer Science and Applications, Kurukshetra University. www.IJCSI.org

[38] Ridha Soua, Arief B. Koesdwiady, and Fakhri Karray. 2016. Big-data-generated traffic flow prediction using deep learning and dempster-shafer theory. *2016 International Joint Conference on Neural Networks (IJCNN)* (2016), 3195–3202.

[39] Guanting Tang, Jian Pei, James Bailey, and Guozhu Dong. 2015. Mining multidimensional contextual outliers from categorical relational data. *Intelligent Data Analysis* 19, 5 (9 2015), 1171–1192. https://doi.org/10.3233/IDA-150764

[40] Nicolas Tempelmeier, Yannick Rietz, Iryna Lishchuk, Tina Kruegel, Olaf Mumm, Vanessa Miriam Carlow, Stefan Dietze, and Elena Demidova. 2019. Data4UrbanMobility: Towards Holistic Data Analytics for Mobility Applications in Urban Regions. In *Companion Proceedings of The 2019 World Wide Web Conference (WWW '19)*. ACM, New York, NY, USA, 137–145. https://doi.org/10.1145/3308560.3317055

[41] Xiang Wang and Ian Davidson. 2009. *Discovering Contexts and Contextual Outliers Using Random Walks in Graphs*. Technical Report. Ninth IEEE International Conference on Data Mining.

[42] W.K. Wong, Min Xia, and W.C. Chu. 2010. Adaptive neural network model for time-series forecasting. *European Journal of Operational Research* 207, 2 (12 2010), 807–816. https://doi.org/10.1016/J.EJOR.2010.05.022

[43] Y. Xia, X. Cao, F. Wen, G. Hua, and J. Sun. 2015. Learning Discriminative Reconstructions for Unsupervised Outlier Removal. In *2015 IEEE International Conference on Computer Vision (ICCV)*. 1511–1519. https://doi.org/10.1109/ICCV.2015.177

[44] Ming Yang and X Rong Li. 2003. *Predicting End-to-End Delay of the Internet Using Time Series Analysis **. Technical Report. University of New Orleans, New Orleans. https://pdfs.semanticscholar.org/ea19/05c52e462560783715a1778c9d4c12352504.pdf

[45] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu. 2017. LSTM network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems* 11, 2 (2017), 68–75. https://doi.org/10.1049/iet-its.2016.0208

[46] Eric R. Ziegel. 1995. Time Series Analysis, Forecasting, and Control. *Technometrics* 37, 2 (5 1995), 238–242. https://doi.org/10.1080/00401706.1995.10484323