Valentina Đorđević*, Pavle Milošević, Ana Poledica
University of Belgrade, Faculty of Organizational Sciences, Serbia

Machine Learning Based Anomaly Detection as an Emerging Trend in Telecommunications

DOI: 10.7595/management.fon.2020.0002

Abstract:

Research Question: This paper investigates into how machine learning can be applied for the purpose of detecting anomalies in the data describing transport component within the cellular network. **Motivation**: In the field of telecommunications, terabytes of data are generated each hour. This makes the manual analysis almost impossible to perform. There are thousands of components whose behaviour needs to be monitored, since anomalous behaviour could indicate a possible failure that can lead to network degradation, huge maintenance costs, and finally – a bad user experience. Our goal is to try to catch anomalous behaviour automatically, and thus help domain experts when performing drill down analysis of the degraded base stations and their key performance indicators (KPIs). **Idea**: The main idea of this paper is to empirically evaluate the application of machine learning for the problem of anomaly detection, in the field of telecommunications, specifically to long term evolution (LTE) networks. **Data**: Data used in the analysis contains information about base transceiver stations (BTS) behaviour through the time. The data are gathered from a cellular network provider located in Serbia. The data are collected on an hourly basis, for a period of two weeks, resulting in almost 700 thousand rows. The behaviour is assessed by 96 transport KPIs coming from BTS, describing the package losses, delays, transmission success rates, etc. **Tools**: Two main algorithms, ensemble-based Isolation Forest and autoencoder neural network, are elaborated and applied in order to identify patterns of anomalous behaviour. **Findings**: The results show that machine learning can be successfully applied in the field of LTE networks for the problem of anomaly detection. Machine learning can significantly reduce the time needed for the domain experts to identify anomalies within the network. In addition to time efficiency, one of the algorithms tested is able to identify anomalous KPIs separately, which is crucial when performing root cause analysis, by using drill-down approach, in order to identify which component is degraded. **Contribution**: This paper enriches existing research related to anomaly detection in LTE networks and provides an innovative approach to automated root-cause analysis of network degradation.

Keywords: anomaly detection, LTE network, telecommunications, autoencoder, isolation forest, machine learning

JEL Classification: C63, L96

1. Introduction

Anomalies are patterns in data that do not conform to a well-defined notion of normal behaviour (Chandola, Banerjee, & Kumar, 2009). Anomaly detection is a task of identifying instances whose behaviour significantly differs from normal or some expected behaviour in the data. The importance of anomaly detection can be seen in identification of faults (Lin, Ke, & Tsai, 2015), failures (Chernogorov, Ristaniemi, Brigatti, & Chernov, 2013; Chernov, Cochez, & Ristaniemi, 2015; Mueller, Kaschub, Blankenhorn, & Wanke, 2008), or frauds (Yee, Sagadevan, & Malim, 2018) in the systems being modelled.

* Corresponding author: Valentina Đorđević, e-mail: valentinadj4@gmail.com

In most cases, the completeness of the data directs the flow of the analysis. The information of anomalous behaviour is often not known or expensive to obtain, and in that case, only normal behaviour is modelled. When dealing with anomalies in the data, it is often needed to observe contextual features (such as time and space), among behavioural features that are usually analysed. Many different techniques and approaches have been developed in order to deal with anomalies in the data, tailor-made to respond to different challenges such as data incompleteness, domain specificity, resource utilization and scalability.

Anomaly detection has been applied in a variety of ways in the field of telecommunications, especially in long term evolution (LTE) networks analysis. In most cases, key performance measurements and parameters from radio and transport network have been analysed in order to identify anomalous behaviour. Many different approaches, such as self-organizing networks (Mueller, Kaschub, Blankenhorn, & Wanke, 2008), ensemble methods (Chernogorov, Chernov, Brigatti, & Ristaniemi, 2016) and knowledge-based methods (Karatepe & Zeydan, 2014) have been tested. However, most of them have not been applied in the business environment.

Machine learning is the field of study that gives computers the ability to learn hidden patterns and internal structure of some system, without being explicitly programmed (Samuel, 1959). During the past decade, machine learning techniques have been widely used for different kinds of problems, including anomaly detection (Purarjomandlangrudi, Ghapanchi, & Esmalifalak, 2014). The main advantages of machine learning techniques in the field of anomaly detection are efficiency and automation. Machine learning techniques are developed to handle unlabelled data, multivariate time-series data, scalable data and incomplete data (Alpaydin, 2014).

LTE networks are a very attractive field for machine learning applications. Telecommunications are one of the fastest growing areas, since enormous amounts of data are generated each hour, and there are thousands of key performance measures (KPIs) and configuration parameters that should be tracked in order to analyse network behaviour and diagnose problems. It is almost impossible for transport and radio experts to analyse all these data and identify all base transceiver stations (BTS) in the LTE network that have anomalous behaviour. This process is nowadays mostly done by reduction of relevant KPIs, where experts track only several main KPIs and determine of anomalous behaviour based on that subset only. Machine learning makes it possible to automatically analyse thousands of BTS with hundreds of their behavioural and contextual features reflected in KPIs, to identify the BTS with deviant behaviour.

Most interesting types of applications of machine learning in the LTE network analysis include anomaly detection and maintenance prediction. These two problems are the most challenging ones, and directly affect network performances and later – the user experience. The problems of sleeping cells and cell outages are analysed in detail (Chernogorov et al., 2013; Chernogorov et al., 2016; Mueller et al. 2008). In (Wu, Lee, Li, Pan, & Zhang, 2018), anomalies defined as sudden drops and correlation changes of KPIs in a LTE network are identified via regression based anomaly detection. The application of machine learning for smart next-generation wireless networks (5G) is presented in (Jiang et al. 2017; Hammami, Moungla, & Afifi, 2018; Polese et al., 2018).

In this paper, we aim to develop two different models for the purpose of anomaly detection in LTE networks regarding the transport component of the network. The main task is to compare their performances and choose the model which is more applicable in case of the given problem. The problem includes identification of deviant values for a given set of KPIs per BTS, describing the transport performances. It is important to identify these deviations regarding all relevant KPIs simultaneously, because their interdependencies also matter. Anomaly is reflected in:

- degradation of one KPI, while other KPIs have expected values;
- increase in one KPI, while other KPIs have expected values;
- multiple KPIs have degradations or increase which indicates a problem on the network.

The proposed model should be able to automatically identify anomalous behaviour in the LTE network. It should determine which features are degraded and what the intensity of their degradation is, indicating a possible root cause of the deviant behaviour. The data used as an input come from the radio network, and contain the BTS behaviour over time. This behaviour is reflected in key performance measures such as package delays, package losses, message transmission success rates, etc. The main characteristic of the model is that it enables continuous health checks, efficient problem diagnosis, and process automation. These beneficial characteristics are of high value for the business, since they could directly impact the process of decision making and network optimization, which can result in an enormous reduction of costs. We believe that this paper has a potential of showing how machine learning can ease and improve business

processes and thus condition the business transformation. The proposed solution has been tested on real-world data, and the results of the application are presented in the paper.

This paper is structured as follows. In Section 2 we cover the basic concepts of anomaly detection. Section 3 is dedicated to introducing the machine learning as alternative approach to anomaly detection, where two main algorithms are elaborated: ensemble-based Isolation Forest and autoencoder neural network. In Section 4 we present the case study, along with the problem definition and main requirements. Section 5 contains main conclusions driven from the analysis, as well as directions for further research that are to be taken.

2. Anomaly Detection

Anomaly detection, as the field of study, has been researched within a wide range of application domains and underlying scientific approaches. The application varies from cyber-intrusions, credit card frauds and system breakdowns, to components faults and parameter misconfiguration. The application domains are also numerous, from banking, aviation and telecommunications, to insurance companies and software development.

Regarding the CRISP-DM methodology, the process of anomaly detection usually starts with understanding the main concepts of the domain being analysed by Chapman et al. (2000). Further, it is necessary to cooperate with the domain experts, and to define the anomalous behaviour that should be identified and tracked. The analysis starts with descriptive statistics that helps identifying deviant behaviour and dimensions containing deviations on a high level of abstraction. After this step, it is determined whether the clustering of the data is needed prior to the identification of anomalies. For example, in a telecommunication network, there are several thousand cells, with different configurations and performances, and they are usually divided by the channel bandwidth, so the anomaly detection analysis should be conducted separately for different groups of cells. After the optional clustering process is done, the appropriate anomaly detection algorithm is applied. The most important step in this process is setting the optimal parameter values, which often includes consulting the domain expert, detailed research of the algorithm and the automation of parameter tuning. After the parameter tuning is done and optimal values are set, the algorithm is tested and evaluated against a new set of data and optionally - other anomaly detection algorithms. When the appropriate algorithm is chosen and applied, the final step includes interpreting the anomalies, drawing conclusions and defining future actions that should reduce or prevent anomalous behaviour from happening again.

Anomaly detection techniques are numerous and versatile, but they can be classified into several groups, based on the data being available and the underlying mechanism used for anomaly score calculation. There are plenty of other classifications, which can be found in (Agrawal & Agrawal, 2015; Chandola et al., 2009; Hodge & Austin, 2004).

Anomaly detection techniques based on the data can be grouped as follows:

1. supervised techniques - includes modelling both the normal and anomalous behaviour. It is analogical to supervised approach for classification problem, and it requires labelled data.
2. unsupervised techniques - searching for anomalies with no previous knowledge of the data. It is analogical to unsupervised approach used for clustering, where similar instances are grouped into clusters, based on some similarity measure - whether it be distance, density or the position of the assigned node in a binary tree.
3. semi-supervised techniques - a mixture of the previous two types. This approach includes modelling of just one type of behaviour, the most frequent – a normal one. It is considered to be semi-supervised since the model learns over the instances belonging to only one class.

Anomaly detection techniques based on the probability score calculation are of high variety, from probability and distance based, to reconstruction and isolation based techniques. A complete overview on the anomaly detection techniques can be found in the survey presented in (Chandola et al., 2009).

3. Machine Learning Techniques for Anomaly Detection

Machine learning techniques are widely used in the anomaly detection problem. Some of the most common approaches are explained in a survey presented in (Agrawal & Agrawal, 2015). Machine learning techniques are most frequently used when there is a need to work with multivariate, unlabelled data, containing lots of

noise. Machine learning can be applied for anomaly detection in a variety of manners. Some of the most popular cases include intrusion detection (Buczak & Guven, 2016; Lin et al., 2015), credit card frauds (Yee et al., 2018), and time series anomaly detection (Malhotra, Vig, Shroff, & Agarwal, 2015; Radford, Apolonio, Trias, & Simpson, 2018). For the purpose of anomaly detection, numerous different approaches were taken, e.g., artificial neural networks (Schmidhuber, 2015) and Bayesian networks (Friedman, Geiger, & Goldszmidt, 1997). Another interesting approach includes using forecasting models such as ARIMA, as presented in (Yu, Jibin, & Jiang, 2016). An example of anomaly detection by using clustering techniques is studied in (Lin et al., 2015).

In this paper, we analyse two different types of algorithms, ensemble-based Isolation Forest, and autoencoder neural networks. Isolation forest, as an ensemble-based method, is used as a baseline model, since it is efficient and has already been used by the authors in other business user cases. Due to its limitations, and as regards the superiority and expansion of neural networks in the past years, the other method is chosen as another approach. The main idea is to compare their performances and determine their convenience for the given problem of transport anomaly detection in the LTE network.

3.1 Isolation forest anomaly detection

Isolation forest (iForest) is a machine learning algorithm coming from a group of ensemble-based methods (Liu, Ting, & Zhou, 2008). It runs in linear time and it is able to work with high-dimensional data with redundant features. The iForest algorithm approaches the problem of anomaly detection by isolating anomalous instances, while most of other anomaly detection algorithms are actually focusing on the profiling of normal behaviour, where the anomalies are interpreted as deviations from that normal profile. The iForest algorithm essentially works by generating a set of trees, called isolation trees (iTrees) for a given set of data, after which the anomalies are determined by choosing the instances with the shortest average path length within the iTree. The main idea is that anomalies should be easily isolated, since they are highly deviant and rare. The anomaly score and path length are inversely proportional, thus the shorter the path, the higher is the anomaly score for a given instance, and vice versa.

Since the iForest algorithm can work both in supervised and unsupervised modes, it works in two phases (Liu et al., 2008):

1. training phase - construction of isolation trees by random selection of sub-samples;
2. testing phase - passing test instances through iTrees to obtain an anomaly score for each instance.

In the training phase, iTrees are constructed by recursive partitioning of the given training set until instances are isolated or a specific tree height is reached, which results in a partial model. In the testing phase, an anomaly score is derived from the expected path length for each test instance. More details regarding the construction of the tree and the algorithm itself can be found in (Liu et al., 2008). The path lengths are derived by passing instances through each iTree. A simplified representation of an iForest is given in Figure 1.

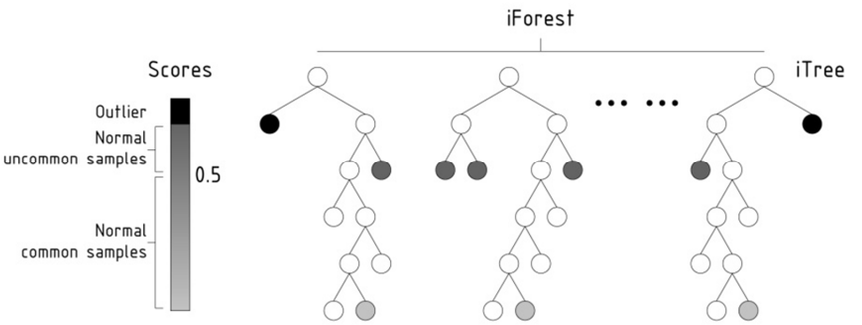

Figure 1: Isolation forest algorithm

There are two basic parameters within this method: the number of trees that will form the set and the size of the subsamples for the training. This algorithm quickly converges with a small number of trees and a small subgroup of data to achieve a high performance detection of anomalies and a high efficiency of execution. One way to detect anomalies is to sort data points $x_1,...,x_n$ according to their path lengths $h(x)$ or anomaly scores; and anomalies are points that are ranked at the top of the list (Liu et al., 2008).

Technically, the isolation score $s(x,n)$ is derived from the average path length $c(n)$ (Liu et al., 2008). Since the structure of the iTree is equivalent to the structure of a binary search tree (BST), the estimation of the average path length $c(n)$ is analogical to the length of unsuccessful search in the BST. The average path of the unsuccessful search in the BST is calculated as:

$$c(n) = 2 \cdot H(n-1) - \frac{2 \cdot (n-1)}{n}, \tag{1}$$

where $H(i)$ is the harmonic number and it can be estimated by $\ln(i)$ + Euler's constant.

The anomaly score s of an instance x is defined as:

$$s(x,n) = 2^{-\frac{E(h(x))}{c(n)}}, \tag{2}$$

where $E(h(x))$ is the average of $h(x)$ from a collection of isolation trees. From the Equation (2), the following can be concluded:

- when $E(h(x)) \to c(n), s \to 0.5$;
- when $E(h(x)) \to 0, s \to 1$;
- when $E(h(x)) \to n-1, s \to 0$.

Using the anomaly score s, the following assessments could be made:

- if instances return s very close to 1, then they are definitely anomalies,
- if instances have s much smaller than 0.5, then they are quite safe to be regarded as normal instances, and
- if all the instances return $s \approx 0.5$, then the entire sample does not really have any distinct anomaly.

Figure 2 presents a relationship between expected path length and anomaly score. The anomaly score can be seen as decreasing with the increase in the expected path length. Further information regarding the iForest algorithm can be found in (Liu et al., 2008).

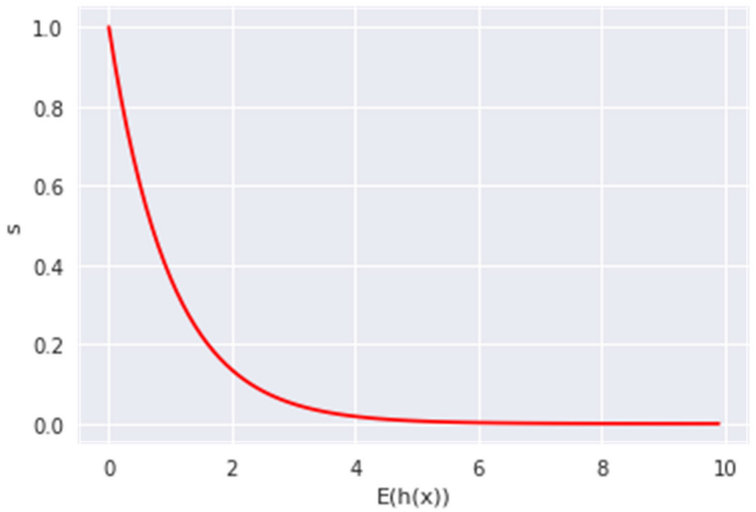

Figure 2: The relationship between the anomaly score (s) and the path length (E(h(x)))

As previously mentioned, the main advantage of the iForest algorithm is that it can work both in supervised and unsupervised modes. It is a fast and efficient algorithm (Liu, Ting, & Zhou, 2012), although it has some shortcomings (e.g., inability to work with multivariate time series). The application of iForest algorithm has been presented in various case studies, covering different problem domains such as streaming data analysis (Ding & Fei, 2013) and in log analysis (Sun, Versteeg, Boztas, & Rao, 2016).

3.2 Autoencoder anomaly detection

Artificial neural networks (ANNs) are computing models developed to handle complex, non-linear relationships and hidden structures in the data (Yegnanarayana, 2009). ANN consists of interconnected processing units, called neurons. A simple representation of a neuron is depicted in Figure 3. The main task of a neuron is to process a certain input vector, perform some computations over it, and generate an output. Thus, the general model of a neuron consists of a summing part (yellow area) followed by an output part (blue area). The summing part receives *N* input values, weights each value, and computes a weighted sum. The weighted sum in this context is called the activation value. The output part produces a signal from the activation value, by passing it through a chosen, usually non-linear, activation function.

An artificial neuron can be mathematically represented by using the formula:

$$y = \varphi\left(\sum_{i=0}^{m} w_i x_i + b\right), \tag{3}$$

where x is the input vector represented $x = [x_1, x_2, ..., x_m, i = 1..m,\ m$ is the number of input features, w is the vector of weights represented as $w = [w_1, w_2, ..., w_n$ joined to each of the input values from the input vector, φ is the activation function used for transformation, and y is the output value generated as a result of this transformation. b represents the bias which is a constant used for getting the best fit over the data, given the input vector.

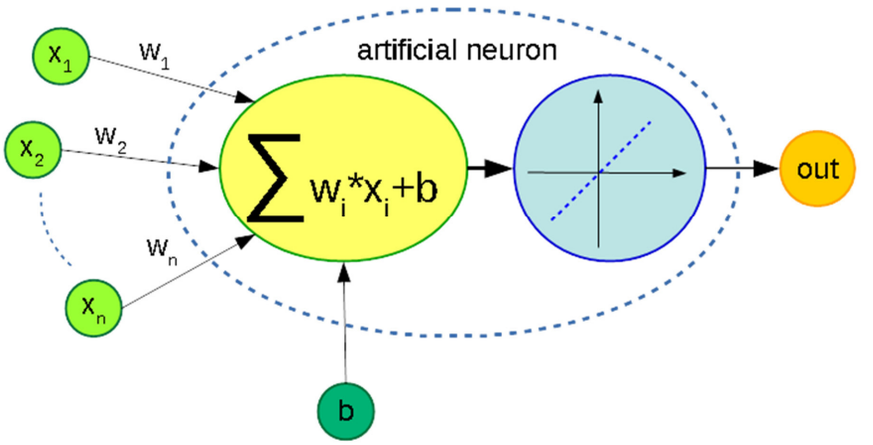

Figure 3: An artificial neuron

ANN generally consists of multiple interconnected layers, each containing an arbitrary number of neurons. It can be noticed that two layers are mandatory - an input layer, used to take inputs for computation, and an output layer that generates computation results. The number of neurons in the input layer depends on the number of features that are given as inputs. The number of neurons in the output layer depends on the number, or more precisely - the type - of an output that should be generated. Layers between the input and output layer are in charge of discovering the hidden structure in the data, and those layers are called hidden layers. The number of hidden layers and belonging units varies, which results in numerous architectures and network topologies, developed to handle different machine learning problems. For further information, consider reading overview presented by Schmidhuber (2015).

The Autoencoder is a special architecture of a neural network that works in an unsupervised mode (An & Cho, 2015; Yegnanarayana, 2009; Vincent, Larochelle, Lajoie, Bengio, & Manzagol, 2010). A simplified architecture is presented in Figure 4.

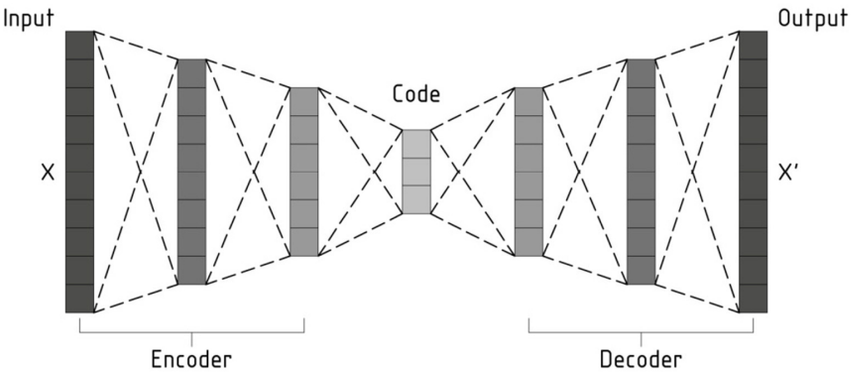

Figure 4: Autoencoder architecture

The main task of an autoencoder is to learn a pair of non-linear transformations (Baldi, 2012):

1. encoder - mapping an input *X* from the original space to another space, but optionally of higher or lower dimensionality;
2. decoder - mapping an encoded input to its original space *X'*.

The idea is, since anomalies are highly deviant and rare, that the autoencoder will not be able to learn those anomalous instances correctly, which results in a higher reconstruction error. A typical definition of the reconstruction error is given as follows:

$$S = \sum_{i=1}^{N} \left| y_i - \varphi_{(x_i)} \right| \tag{4}$$

Another common approach is to calculate the reconstruction error by using the cross-entropy loss function:

$$L\left(X, X^{'}\right) = \sum_{c=1}^{N} X_c^{'} \cdot \log\left(X_c\right) \tag{5}$$

The Autoencoder introduces the reconstruction error as an anomaly score for a given instance. A fraction of instances with a highest reconstruction error are considered anomalous. The size of this fraction can vary, and it depends on the domain being analysed.

Traditional autoencoder and some modern modifications like variational autoencoder or mutual autoencoder are widely used in different fields, e.g., for anomaly detection in aviation (Sakurada & Yairi, 2014), automation (Park, Hoshi, & Kemp, 2018), intrusion detection (Lopez-Martin, Carro, Sanchez-Esguevillas, & Lloret, 2017), etc.

4. Experiment

The data mining problem that we are focusing on is the detection of anomalies on the transport part of the LTE network, by analysing transport KPIs coming from BTS. We are approaching this problem regarding the life cycle of a data mining project given by the CRISP-DM reference model (Chapman et al., 2000). The CRISP-DM is highly compliant with the agile approach, often used in IT projects. The first phase in this model is business understanding, which focuses on understanding the project objectives and requirements from a business perspective. This knowledge is then converted into a data mining problem definition and a preliminary plan is designed to achieve the objectives. The second phase is data understanding, where the goal is to become familiar with the data, in order to determine its predictive power, as well as its constraints. The following steps are data preparation and modelling, where data mining algorithms are applied in order to solve the problem. In the evaluation phase that follows, the results are evaluated, and if they are satisfying, they go to the deployment phase. Otherwise, iteration through previous phases is done until satisfying results are not reached.

4.1 Problem setup

Business understanding. A telecommunication network is a collection of BTS, interconnected by transmission links that enable communication between two distant entities. BTS are connected to other BTS or hubs via microwave links, while hubs are connected to the network core via optical links. The network core is the brain of a telecommunication network, and that is where all the requests are processed. While the message is being transmitted, it passes through several entities until it is being processed in the network core. Thus, there are several problems that might occur. The one we are giving a focus on in this paper is anomalous behaviour of the BTS regarding the transportation of packages. When the BTS does not work properly, the user encounters a problem with the connection, which results in bad user experience. The idea is to detect anomalous behaviour of the BTS, in order to analyse the root causes, and to define some actions that should prevent it from happening again. The main goal is to reduce faults, congestions and user complaints, as well as to improve the overall user experience.

The project objective of this case study is to identify transport anomalies on BTS. The main requirement is to enable this process to be carried out automatically. The data mining problem is defined as unsupervised anomaly detection.

Data understanding. The data available for the analysis contain information about BTS behaviour through time. They are gathered from a cellular network provider located in Serbia. The data have been collected on an hourly level, for a period of two weeks, resulting in almost 700 thousand rows. The behaviour is given by 96 transport KPIs coming from BTS, describing the package losses, delays, transmission success rates, etc. Since labels indicating anomalous behaviour are not available, the choice of machine learning algorithms has been reduced to the group of unsupervised algorithms.

Data preparation. The data preparation phase is dedicated to creating a structure of the data convenient for analytical purposes. The data are transformed into a matrix form, where rows are BTS/hour instances, while columns represent KPIs values for each instance. Testing hours are excluded from the analysis, in order to prevent drawing wrong conclusions. Missing values have been imputed using regression-based technique presented in (White, Royston, & Wood, 2011). The iForest algorithm does not require additional feature normalization, while for the purpose of training the autoencoder, the data normalization has been performed. This has been done in order to assure that all KPIs are of the same importance.

Modelling. The modelling phase includes building and applying various modelling techniques, tuning up the model parameters and obtaining model results. In this phase, two models elaborated in previous sections are applied – the Isolation Forest and autoencoder. The first step in the modelling phase includes training the iForest algorithm as a baseline model. The second step is dedicated to training the autoencoder as an alternative algorithm. The identical set of instances with their features has been passed to both models as an input. The output of this phase is the anomaly score for each instance.

4.2 Results and Discussion

Evaluation. Performing the evaluation is always challenging when using unsupervised algorithms, but it is necessary in order to determine whether the models developed are able to meet the project objectives. One approach taken here is the evaluation performed by the domain experts. Since it is impossible to evaluate each instance separately, another approach has been developed for labelling the data. In this approach, ten major KPIs are filtered out of the KPIs used in the modelling phase. Based on the defined thresholds for these KPIs, each instance is flagged as normal or anomalous. The output from the modelling phase is transformed in such way that a certain percentage of instances with the highest anomaly score is flagged as anomalous, while the rest is flagged as normal. This percentage is determined by the domain experts. Since we have labelled the data, we can calculate accuracy score and perform rough evaluation of the models. False positive and false negative rates are also a popular metrics for this kind of problem (Nakayama, Kurosawa, Jamalipour, Nemoto, & Kato, 2008). Regardless of their significance, we have not presented them because they are of the same order of magnitude and had no effect on the algorithm selection in our case. However, we briefly discuss the importance of the individual false positive observation from the aspect of potential costs. Furthermore, we reflect on transparency of the results and the possibility of preparing corrective actions when certain anomalies are detected.

The table that follows shows a quantitative analysis, as well as main advantages and disadvantages of both approaches. It can be noticed that the iForest algorithm has a higher accuracy, but it is not accuracy that

only matters. It is more important to extract the KPIs that are anomalous, for each instance separately. Although the iForest is more efficient and faster, the autoencoder gives the output that is more valuable, regarding the interpretation of anomalous features and conclusion drawing. That is why the autoencoder is used as a more convenient approach in comparison with the baseline iForest algorithm. The summary is given in Table 1, and the main findings are discussed below.

This analysis proves that both algorithms are able to identify anomalies in the network with an accuracy of about 70%. The rates of false positive and false negative observations are of the same order of magnitude for either analysed algorithm. It should be noticed that the obtained results are incomparable with other studies in general, since anomaly detection accuracy highly depends on data. Also, our dataset and labelling method are rather specific. Still, in literature (Ahmed, Mahmood, & Hu, 2016; Krömer, Platoš, Snášel, & Abraham, 2011), the accuracy of different algorithms for anomaly detection oscillates from 57% to 82%. Therefore, it can be said that our results are in line with the state-of-the-art approaches.

Table 1: Comparative and quantitative analysis of applied algorithms

	Isolation Forest	**Autoencoder**
Accuracy	**72%**	**69%**
Advantages	- Very fast and efficient - Good performances with redundant data - Works both in supervised and unsupervised mode	- Gives anomaly score for each feature - Good with catching non-linear dependencies - Convenient for noise reduction
Disadvantages	- Not possible to extract anomaly score for each feature - Not possible to visualise isolation trees	- Takes too much time with high-dimensional data - Tries to capture as much information as possible, rather than as important information as possible

Isolation Forest has shown to be extremely fast, requiring minimal parameter tweaking. However, one of the biggest drawbacks is that it does not return anomaly score per each KPI separately. That seriously inflicts the transparency of the obtained results and it is unpopular among a majority of non-technical decision makers. Also, the absence of individual anomaly scores limits the potential for preparing and undertaking corrective actions for specific anomalies. Nevertheless, we believe that this limitation will soon be overcome, thus the real advantages of this algorithm among others will be clearly separated.

Regarding the autoencoder, we have had memory and time consummation issues. Besides, it has a lot more hyper parameter tuning, and thus is more complex than the previous one. On the other hand, the autoencoder is able to extract anomaly score for each KPI separately. Besides that, the anomaly score is also directly positively correlated to the level of degradation, and this is the main information we wanted to gain as insight. This insight is of a very high value, because it gives us the opportunity not only to highlight anomalous BTS and KPIs, but also to rank KPIs and extract most anomalous ones. When the information of most anomalous KPIs is provided, domain experts can easily determine the root causes and define healing actions.

One of the most important user requirements was to analyse all features simultaneously, and it was almost impossible to define thresholds for each KPI separately. That is the main reason why the unsupervised approach has been taken. We wanted to let the algorithm determine when each of the KPIs is anomalous. The highest benefit could be seen in automated identification of anomalous KPIs in a bunch of other KPIs, which was extremely time-consuming for the domain experts when performing manual analysis.

Since the accuracies of the algorithms are rather even and the autoencoder is able to give anomaly score for each KPI separately, this algorithm is chosen as a more convenient approach for the problem being defined. The autoencoder's ability to rank KPIs from normal to the most anomalous ones has been proved to be very important in practice. This has led to prioritization of KPIs in order to ease network analysis, i.e. simplification and speeding up the examination of degraded parameters and problematic networks configurations in the real world application. Furthermore, it enables identifying possible strong causal relationships between anomalous KPIs and alarms in the network, e.g., a degradation of established connections per-

centage per cell and a complete failure of the cell. Finally, for the most of detected anomalies, network operators or radio experts can define automatic healing procedures that would resolve the problem in a swift and easy manner. Therefore, the proposed anomaly detection procedure would significantly improve performance of the network and reduce maintenance costs. Still, some anomalies that require on-site solutions are detected with relatively high number of false positives. Although false positive observations are usually of the particular interest for anomaly detection, in these cases it is an open question of whether the operator should send a team on site due to costs of a possible 'false alarm'.

Conclusion

Using machine learning for process automation and efficiency achievement has been proven as a useful approach with real-world application and remarkable benefits. It shows outstanding performances in solving problems containing vast amounts of data. In this paper, the machine learning techniques are used as a tool for automated anomaly detection in a specific field – LTE cellular network. The data used for analysis contained transport KPIs describing the behaviour of the base stations regarding the transport capabilities.

Two approaches have been analysed – an ensemble-based Isolation Forest algorithm and a neural network based autoencoder. Both approaches have certain fortes. The iForest algorithm is fast and efficient, while the autoencoder is noise-proof and calculates a particular feature anomaly score. The main and the most valuable advantage of the autoencoder is its ability to return the anomaly score for each KPI. That is why the autoencoder is chosen as more favourable over the iForest.

The future work will include training the models over the data containing more history, from three to six months. One of the biggest challenges will be the scalability. The idea is to aggregate the data on a daily level, in order to reduce complexity and hourly fluctuations. Further, this analysis can be improved by adding more information regarding other KPIs coming from transport network and network topology. This will improve the accuracy and give more valuable insights into which part of the network is degraded and can be used to find the root causes. Finally, the main idea is to use the results of this analysis as a baseline for translating this problem into the problem of predictive analysis.

Acknowledgements

This research was endorsed by Things Solver, d.o.o., Serbia. We would like to show our gratitude to the Vip Mobile Serbia, which provided us the data for the analysis. We would especially like to thank orđe Begenišić, Radio Expert, Vip Mobile Serbia, whose effective cooperation provided meaningful insights and domain expertise that greatly assisted the research.

REFERENCES

[1] Agrawal, S., & Agrawal, J. (2015). Survey on anomaly detection using data mining techniques. *Procedia Computer Science*, 60, 708-713. DOI: 10.1016/j.procs.2015.08.220
[2] Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60, 19-31. DOI: 10.1016/j.jnca.2015.11.016
[3] Alpaydin, E. (2014). *Introduction to Machine Learning (3rd ed.)*. Cambridge, US: MIT Press.
[4] An, J., & Cho, S. (2015). Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, *2*, 1-18.
[5] Baldi, P. (2012, June). Autoencoders, unsupervised learning, and deep architectures. In I. Guyon, G. Dror, V. Lemaire, G. Taylor & D. Silver (Eds.), *Proceedings of ICML Workshop on Unsupervised and Transfer Learning* (pp. 37-49).
[6] Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, *18*(2), 1153-1176. DOI: 10.1109/COMST.2015.2494502
[7] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, *41*(3), 15. DOI: 10.1145/1541880.1541882
[8] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). CRISP-DM 1.0 Step-by-step data mining guide.
[9] [9] Chernogorov, F., Chernov, S., Brigatti, K., & Ristaniemi, T. (2016). Sequence-based detection of sleeping cell failures in mobile networks. Wireless Networks, 22(6), 2029-2048. DOI: 10.1007/s11276-015-1087-9

[10] Chernogorov, F., Ristaniemi, T., Brigatti, K., & Chernov, S. (2013, May). N-gram analysis for sleeping cell detection in LTE networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing Proceedings* (pp. 4439-4443). IEEE. DOI: 10.1109/ICASSP.2013.6638499
[11] Chernov, S., Cochez, M., & Ristaniemi, T. (2015, May). Anomaly detection algorithms for the sleeping cell detection in LTE networks. In *2015 IEEE 81st Vehicular Technology Conference (VTC Spring) Proceedings* (pp. 1-5). IEEE. DOI: 10.1109/VTCSpring.2015.7145707
[12] Ding, Z., & Fei, M. (2013). An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window. *IFAC Proceedings Volumes*, *46*(20), 12-17. DOI: 10.3182/20130902-3-CN-3020.00044
[13] Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, *29*(2-3), 131-163. DOI: 10.1023/A:1007465528199
[14] Hammami, S. E., Moungla, H., & Afifi, H. (2018, May). Proactive Anomaly Detection Model for eHealth-Enabled Data in Next Generation Cellular Networks. In *2018 IEEE International Conference on Communications (ICC) Proceedings* (pp. 1-6). IEEE. DOI: 10.1109/ICC.2018.8422516
[15] Hodge, V., & Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review*, *22*(2), 85-126. DOI: 10.1023/B:AIRE.0000045502.10941.a9
[16] Jiang, C., Zhang, H., Ren, Y., Han, Z., Chen, K. C., & Hanzo, L. (2017). Machine learning paradigms for next-generation wireless networks. *IEEE Wireless Communications*, *24*(2), 98-105. DOI: 10.1109/MWC.2016.1500356WC
[17] Karatepe, I. A., & Zeydan, E. (2014, May). Anomaly detection in cellular network data using big data analytics. In *Proceedings of European Wireless 2014; 20th European Wireless Conference;* (pp. 1-5). VDE.
[18] Krömer, P., Platoš, J., Snášel, V., & Abraham, A. (2011, October). Fuzzy classification by evolutionary algorithms. In *2011 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 313-318). IEEE. DOI: 10.1109/ICSMC.2011.6083684
[19] Lin, W. C., Ke, S. W., & Tsai, C. F. (2015). CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. *Knowledge-based Systems*, *78*, 13-21. DOI: 10.1016/j.knosys.2015.01.009
[20] Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008, December). Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining* (pp. 413-422). IEEE. DOI: 10.1109/ICDM.2008.17
[21] Liu, F. T., Ting, K. M., & Zhou, Z. H. (2012). Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data*, *6*(1), 3. DOI: 10.1145/2133360.2133363
[22] Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., & Lloret, J. (2017). Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in IoT. *Sensors*, *17*(9), 1967. DOI: 10.3390/s17091967
[23] Malhotra, P., Vig, L., Shroff, G., & Agarwal, P. (2015, April). Long short term memory networks for anomaly detection in time series. In M. Verleysen (Ed.), *Proceedings of ESANN 2015: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* (pp. 89-94). Louvain-la-Neuve, Belgium: Ciaco.
[24] Mueller, C. M., Kaschub, M., Blankenhorn, C., & Wanke, S. (2008, December). A cell outage detection algorithm using neighbor cell list reports. In *International Workshop on Self-Organizing Systems* (pp. 218-229). Heidelberg, Germany: Springer. DOI: 10.1007/978-3-540-92157-8_19
[25] Park, D., Hoshi, Y., & Kemp, C. C. (2018). A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder. *IEEE Robotics and Automation Letters*, *3*(3), 1544-1551. DOI: 10.1109/LRA.2018.2801475
[26] Polese, M., Jana, R., Kounev, V., Zhang, K., Deb, S., & Zorzi, M. (2018). Machine Learning at the Edge: A Data-Driven Architecture with Applications to 5G Cellular Networks. *arXiv preprint arXiv:1808.07647*.
[27] Purarjomandlangrudi, A., Ghapanchi, A. H., & Esmalifalak, M. (2014). A data mining approach for fault diagnosis: An application of anomaly detection algorithm. *Measurement*, *55*, 343-352. DOI: 10.1016/j.measurement.2014.05.029
[28] Nakayama, H., Kurosawa, S., Jamalipour, A., Nemoto, Y., & Kato, N. (2008). A dynamic anomaly detection scheme for AODV-based mobile ad hoc networks. *IEEE Transactions on Vehicular Technology*, 58(5), 2471-2481. DOI: 10.1109/TVT.2008.2010049
[29] Radford, B. J., Apolonio, L. M., Trias, A. J., & Simpson, J. A. (2018). Network Traffic Anomaly Detection Using Recurrent Neural Networks. *arXiv preprint arXiv:1803.10769*.
[30] Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3), 210-229. DOI: 10.1147/rd.33.0210
[31] Sakurada, M., & Yairi, T. (2014, December). Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis* (p. 4-11). ACM. DOI: 10.1145/2689746.2689747
[32] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85-117. DOI: 10.1016/j.neunet.2014.09.003

[33] Sun, L., Versteeg, S., Boztas, S., & Rao, A. (2016). Detecting anomalous user behavior using an extended isolation forest algorithm: an enterprise case study. *arXiv preprint arXiv:1609.06676*.
[34] Yegnanarayana, B. (2009). *Artificial neural networks*. New Delhi, India: Prentice-Hall of India.
[35] Yee, O. S., Sagadevan, S., & Malim, N. H. A. H. (2018). Credit Card Fraud Detection Using Machine Learning As Data Mining Technique. *Journal of Telecommunication, Electronic and Computer Engineering*, *10*(1-4), 23-27.
[36] Yu, Q., Jibin, L., & Jiang, L. (2016). An improved ARIMA-based traffic anomaly detection algorithm for wireless sensor networks. *International Journal of Distributed Sensor Networks*, 12(1), 9653230. DOI: 10.1155/2016/9653230
[37] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P. A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, *11*(Dec), 3371-3408.
[38] White, I. R., Royston, P., & Wood, A. M. (2011). Multiple imputation using chained equations: issues and guidance for practice. *Statistics in Medicine*, *30*(4), 377-399. DOI: 10.1002/sim.4067
[39] Wu, J., Lee, P. P., Li, Q., Pan, L., & Zhang, J. (2018, May). CellPAD: Detecting Performance Anomalies in Cellular Networks via Regression Analysis. In *2018 IFIP Networking Conference (IFIP Networking) and Workshops Proceedings* (pp. 1-9). IEEE. DOI: 10.23919/IFIPNetworking.2018.8697027

Received: 2019-03-12
Revision requested: 2019-05-07
Revised: 2020-01-13 (2 revisions)
Accepted: 2020-02-06

About the Authors

Valentina Đorđević
University of Belgrade, Faculty of Organizational Sciences, Serbia
e-mail: valentinadj4@gmail.com

Valentina Đorđević is a Data Scientist at Things Solver, and a Master of Science in Business Intelligence, at the University of Belgrade, Faculty of Organizational Sciences. Her major professional interests include data science, machine learning, anomaly detection and time series analysis.

Pavle Milošević
University of Belgrade, Faculty of Organizational Sciences, Serbia
e-mail: pavle.milosevic@fon.bg.ac.rs

Pavle Milošević, PhD, is an Assistant Professor at the University of Belgrade, Faculty of Organizational Sciences. He is the author and co-author of more than 50 journal articles, chapters and conference papers. His major professional interests include: computational intelligence, systems theory and control, machine learning, metaheuristics and time series analysis.

Ana Poledica
University of Belgrade, Faculty of Organizational Sciences, Serbia
e-mail: ana.poledica@fon.bg.ac.rs

Ana Poledica, PhD, is an Assistant Professor at the University of Belgrade, Faculty of Organizational Sciences. She is the author and co-author of more than 40 journal articles, chapters and conference papers. Her major professional interests include: computational intelligence, systems theory and control, machine learning, quantitative finance and time series analysis.