Transfer learning for time series anomaly detection

Vincent Vercruyssen, Wannes Meert, and Jesse Davis

Dept. of Computer Science, KU Leuven, Belgium
firstname.lastname@cs.kuleuven.be

Abstract. Currently, time series anomaly detection is attracting significant interest. This is especially true in industry, where companies continuously monitor all aspects of production processes using various sensors. In this context, methods that automatically detect anomalous behavior in the collected data could have a large impact. Unfortunately, for a variety of reasons, it is often difficult to collect large labeled data sets for anomaly detection problems. Typically, only a few data sets will contain labeled data, and each of these will only have a very small number of labeled examples. This makes it difficult to treat anomaly detection as a supervised learning problem. In this paper, we explore using transfer learning in a time-series anomaly detection setting. Our algorithm attempts to transfer labeled examples from a source domain to a target domain where no labels are available. The approach leverages the insight that anomalies are infrequent and unexpected to decide whether or not to transfer a labeled instance to the target domain. Once the transfer is complete, we construct a nearest-neighbor classifier in the target domain, with dynamic time warping as the similarity measure. An experimental evaluation on a number of real-world data sets shows that the overall approach is promising, and that it outperforms unsupervised anomaly detection in the target domain.

Keywords: transfer learning; anomaly detection; time series

1 Introduction

Time series data frequently arise in many different scientific and industrial contexts. For instance, companies use a variety of sensors to continuously monitor equipment and natural resources. One relevant use case is developing algorithms that can automatically identify time series that show anomalous behavior. Ideally, anomaly detection could be posed as a supervised learning problem. However, these algorithms require large amounts of labeled training data. Unfortunately, such data is often not available as obtaining expert labels is time-consuming and expensive. Typically, only a small number of labels are known for a limited number of data sets. For example, if a company monitors several similar machines, they may only label events (e.g., shutdown, maintenance...) for a small subset of them.

Transfer learning is an area of research focused on methods that are able to extract information (e.g., labels, knowledge, etc.) from a data set and reapply it in another, different data set. Specifically, the goal of transfer learning is to improve performance on the target domain by leveraging information from a related data set called the source domain [10]. In this paper, we adopt the paradigm of transfer learning for anomaly detection. In our setting, we assume that labeled examples are only available in the source domains, and that there no labeled examples in the target domain. In the example, we utilize the label information available for machine A to help constructing an anomaly detection algorithm for machine B, where no labeled points are available for machine B.

In this paper we study transfer learning in the context of time-series anomaly detection, which has received less attention in transfer learning [1, 6, 10]. Our approach attempts to transfer instances from the source domain to the target domain. It is based on two important and common insights about anomalous data points, namely that they are infrequent and unexpected. We leverage these insights to propose two different ways to identify which source instances should be transferred to the target domain. Finally, we make predictions in the target domain by using 1-nearest neighbors classifier where the transferred instances are the only labeled data points in the target domain. We experimentally evaluate our approach on a large data set adapted from a real-world data set and find that it outperforms an unsupervised approach.

2 Problem statement

We can formally define the task we address in this paper as follows:

Given: One or multiple source domains D_S with source domain data $\{X_S, Y_S\}$, and a target domain D_T with target domain data $\{X_T, Y_T\}$, where the instances $x \in X$ are time series and the labels $y \in Y$ are $\in \{\texttt{anomaly}, \texttt{normal}\}$. Additionally, only partial label information is available in the source domains, and no label information in the target domain.

Do: Learn a model for anomaly detection $f_T(\cdot)$ in the target domain D_T using the knowledge in D_S, and $D_S \neq D_T$.

Both the source and target domain instances are time series. Thus each instance $x = \{(t_1, v_1), \ldots, (t_n, v_n)\}$, where t_i is a time stamp and v_i is a single measurement of the variable of interest v at time t_i. The problem has the following characteristics:

- The joint distributions of source and target domain data, denoted by $p_S(X, Y)$ and $p_T(X, Y)$, are not necessarily equal.
- No labels are known for the target domain, thus $Y_T = \varnothing$. In the source domain, (partial) label information is available.
- The same variable v is monitored in the source and target domain, under possibly different conditions (e.g., the same machine in different factories).
- The number of samples in the D_S and D_T are denoted respectively by $n_S = |X_S|$ and $n_T = |X_T|$, and no restrictions are imposed on them.

- Each time series in D_S or D_T has the same length d.
- The source and target domain instances are randomly sampled from the true underlying distribution.

3 Context and related work

Several flavors of transfer learning distinguish themselves in the way knowledge is transferred between source and target domain. In this paper we employ instance-based transfer learning. The idea is to transfer specific (labeled) instances from the source domain to the target domain in order to improve learning a target predictive function $f_T(\cdot)$ [6]. In the case of anomaly detection, the target function is a classifier that aims to distinguish normal instances from anomalous instances. However, care needs to be taken when selecting which instances to transfer, because transferring all instances could result in degraded performance in the target domain (i.e., negative transfer) [8]. A popular solution is to define a weight for each transferred instance based on the similarity of the source and target domain. The latter is characterized either by the similarity of the marginal probability distributions $p_S(X)$ and $p_T(X)$, and/or the similarity of conditional probability distributions $p_S(Y|X)$ and $p_T(Y|X)$. Various ways of calculating these weights have been proposed [3, 6, 10]. However, the problem outlined in this paper states that $Y_T = \varnothing$, which is a realistic assumption given that in practice labeling is expensive. Hence, we cannot easily calculate $p_T(Y|X)$. Furthermore, even if the marginal distributions are different, it can still be beneficial to transfer specific instances. Consider the following. Since the target task is anomaly detection, one cares for a classifier that robustly characterizes normal behavior. By adding a diverse set of anomalies to the training data of the classifier, the learned decision surfaces will be more restricted, ensuring a decrease of type 2 errors when detecting anomalies in new, unseen data.

The subject of instance-based transfer learning for time series has received less attention in literature. Spiegel recently proposed a mechanism for learning a target classifier using set of unlabeled time series in various source domains, without assuming that source and target domain follow the same generative distribution or even have the same class labels [7]. However, they require a limited set of labels in the target domain, whereas we have $Y_T = \varnothing$.

4 Methodology

In order to learn the model for anomaly detection $f_T(\cdot)$ in the target domain, we transfer labeled instances from different source domains. To avoid situations of negative transfer (e.g., transferring an instance with the label `anomaly` that maps to a normal instance in the target domain), a decision function decides whether to transfer an instance or not. First, we outline the intuitions behind the decision function based on two commonly known characteristics of anomalous instances (Sec. 4.1). Then, we propose two distinct decision functions (Sec. 4.2 and 4.3). Finally, we describe a method for supervised anomaly detection in the target domain based on the transferred instances (Sec. 4.4).

4.1 Instance-based transfer learning for anomaly detection

The literature frequently makes two important observations about anomalous data:

Observation 1 *Anomalies occur infrequently [2].*

Observation 2 *If a model of normal behavior is learned, then anomalies constitute all unexpected behavior that falls outside the boundaries of normal behavior. This implies that it is impossible to predefine every type of anomaly.*

From the first observation we derive the following property:

Property 1 *Given a labeled instance $(x_S, y_S) \in D_S$ and $y_S = \texttt{normal}$. If the probability of the instance under the true target domain distribution $p_T(x_S)$ is high (i.e., the instance is likely to be sampled from the target domain), then the probability that the true label of the instance in the target domain is $\texttt{normal}$, $p_T(y_S = \texttt{normal}|x_S)$ is also high.*

The second observation allows us to derive the reverse property:

Property 2 *Given a labeled instance $(x_S, y_S) \in D_S$ and $y_S = \texttt{anomaly}$. If the probability of the instance under the true target domain distribution $p_T(x_S)$ is low, then the probability that the true label of the instance in the target domain is $\texttt{anomaly}$, $p_T(y_S = \texttt{anomaly}|x_S)$ is high.*

Notice that in the latter property the time series x_S can have any form, while this is not true for the first property, where the form is restricted by the distribution of the target domain data. Given a labeled instance $(x_S, y_S) \in D_S$ that we want to transfer to the target domain, **Property 1** and **Property 2** allow us to make a decision whether to transfer or not. We can formally define a weight associated with x_S which will be high when the transfer makes sense, and low when it will likely cause negative transfer.

$$w_S = \begin{cases} p_T(x_S) & \text{if } y_S = \texttt{normal} \\ 1 - p_T(x_S) & \text{if } y_S = \texttt{anomaly} \end{cases} \tag{1}$$

However, since each time series x_S can be considered as a vector of length d in $\mathbb{R}^d$ (i.e., it consists of a series of numeric values for continuous variable v), the probability of observing exactly x_S under the target domain distribution must be 0. Instead, we calculate the probability of observing a small interval around x_S, such that:

$$p_T(x_S) = \lim_{\Delta I \to 0} \int_{\Delta I} p_T(x_S) dx \tag{2}$$

where ΔI is an infinitesimally small region around x_S in the target domain. This probability is equal to the true density function over the target domain $f_T(x_S)$. Given that the true target domain density is unknown, we need to estimate it from the data X_T. It is shown that this estimate $\hat{f_T}(x_S)$ can be calculated as follows [4]:

$$\hat{f_T}(x_S) = \frac{1}{n_T} \frac{1}{(h_{n_T})^d} \sum_{i=1}^{n_T} K \left(\frac{x_S - x_i}{(h_{n_T})^d} \right) \tag{3}$$

where $K(x)$ is the window function or kernel in the d-dimensional space and $\int_{\mathbb{R}^d} K(x)dx = 1$. The parameter $h_{n_T} > 0$ is the *bandwidth* corresponding to the width of the kernel, and depends on the number of observations n_T. The estimate $\hat{f_T}(x_S)$ converges to the true density $f_T(x_S)$ when there is an infinite number of observations, $n_T \to \infty$, under the assumption that the data X_T are randomly sampled from the true underlying distribution.

4.2 Density-based transfer decision function

For guaranteeing convergence of $\hat{f_T}(x_S)$ to the true density function, the sample size must increase exponentially with the length d of the time series data. The reasoning is clear; high-dimensional spaces are sparsely populated by the available data, making it hard to produce accurate estimates. However, this is often infeasible in practice (gathering data is expensive). For longer time series d is automatically high, that is, if we treat the time series as a vector in $\mathbb{R}^d$. As a practical solution, we propose to reduce the length d of the time series x_S by dividing it into l equal-length subsequences, each with length $m < d$. For every subsequence s in x_S, the density is estimated using Eq. 3 with a Gaussian kernel:

$$\hat{f}_{T,m}(s) = \frac{1}{n_T} \frac{1}{(h_{n_T}\sqrt{2\pi})^m} \sum_{i=1}^{n_T} \exp \left(-\frac{1}{2} \left(\frac{s - s_i}{h_{n_T}} \right)^2 \right) \tag{4}$$

where h_{n_T} is the standard deviation of the Gaussian, and s_i are the subsequences of the instances in X_T. The Gaussian kernel ensures that instead of simply counting similar subsequences, the count is weighted for each subsequence s_i based on the kernelized distance to s_S.

Estimating the densities for the subsequences yields more accurate estimates given the reduced dimensionality, but simultaneously results in $l = m/d$ estimates for each time series x_S. Hence, we have to adjust Eq. 1 to reflect this new situation. We only show the case in which the label $y_S =$ `normal` as the reverse case is straightforward:

$$w_S = \frac{1}{Z_{max} - Z_{min}} \left(\sum_{i=1}^{l} \hat{f_{T,m}}(s_i) - Z_{min} \right) \tag{5}$$

$$Z_{max} = \max_{x_T \in \{X_T \cup x_S\}} \sum_{s_j \in x_T} \hat{f_{T,m}}(s_j) \tag{6}$$

The sum of the density estimates in the subsequences is normalized using min-max normalization, such that $w_S \in [0, 1]$. Z_{min} is calculated similarly as Z_{max} in Eq. 6, but taking the minimum instead of maximum. By setting a threshold on the final weights, we make a decision on whether to transfer or not.

4.3 Cluster-based transfer decision function

Our second proposed decision function is also based on the intuitions outlined in Sec. 4.1. First, the target domain data X_T are clustered using k-means clustering. Second, the resulting set of clusters C over X_T is divided into a set of large clusters, and a set of small clusters according to the following definition [5]:

Definition 1. *Given a dataset X_T with n_T instances, a set of ordered clusters $C = \{C_1, ..., C_k\}$ such that $|C_1| \geq |C_2| \geq ... \geq |C_k|$, and two numeric parameters α and β, the boundary b between large and small clusters is defined such that either of the following conditions holds:*

$$\sum_{i=1}^{b} |C_i| \geq n_T \times \alpha \tag{7}$$

$$\frac{|C_b|}{|C_{b+1}|} \geq \beta \tag{8}$$

$LC = \{C_i | i \leq b\}$ and $SC = \{C_i | i > b\}$ are respectively the set of large and small clusters, and $LC \cup SC = C$.

Furthermore, we define the radius of a cluster as $r_i = \max_{x_j \in C_i} \|x_j - c_i\|^2$. Lastly, a decision is made whether or not to transfer a labeled instance x_S from the source domain. Intuitively, and in line with **Observation 1 and 2**, anomalies in X_T should fall in small clusters, while large clusters contain the normal instances. Transferred labeled instances from the source domain should adhere to the same intuitions. Each transferred instance is assigned to a cluster $C_i \in C$ such that $\|x_S - c_i\|^2$ is minimized. An instance is only transferred in two cases. First, if the instance has label `normal` and is assigned to a cluster C_i such that $C_i \in LC$ and the distance of the instance to the cluster center is less or equal to the radius of the cluster. Second, if the instance has label `anomaly` and fulfills either of two conditions: the instance is assigned to a cluster C_i such that $C_i \notin LC$, or it is assigned to a cluster C_i such that $C_i \in LC$ and the distance of the instance to the cluster center is larger than the radius of the cluster. In all other cases there is no transfer.

4.4 Supervised anomaly detection in a set of time series

After transferring instances from one or multiple source domains to the target domain using the decision functions in Sec. 4.2 and 4.3, we can construct a classifier in the target domain to detect anomalies. Ignoring the unlabeled target domain data, we only use the set of labeled data $L = \{(x_i, y_i)\}_{i=1}^{n_A}$, n_A being the number of instances transferred. It has been shown that one-nearest-neighbor (1NN) classifier with dynamic time warping (DTW) or Euclidean distance is a strong candidate for time series classification [9]. To that end, we construct a 1NN-DTW classifier on top of L to predict the labels of unseen instances.

5 Experimental evaluation

In this section we aim to answer the following research questions:

- Do the proposed decision functions for instance-based time series transfer succeed in transferring useful knowledge between source and target domain.

First, we introduce the unsupervised baseline method to which we will compare the 1NN-DTW method with instance transfer (Sec. 5.1). Then, we discuss the data, the experimental setup, and the results (Sec. 5.2).

5.1 Unsupervised anomaly detection in a set of time series

Without instance transfer, the target domain consists of a set of unlabeled time series data $U = \{(x_i)\}_{i=1}^{n_T}$. Based on the anomaly detection approach outlined in Kha et al., we introduce a straightforward unsupervised algorithm for anomaly detection that will serve as a baseline [5]. The algorithm calculates the *cluster based local outlier factor* (CBLOF) for each series in U.

Definition 2. *Given a set of large LC and small clusters SC defined over U (as per definition 1), the CBLOF of an instance $x_i \in U$, belonging to cluster C_i, is calculated as:*

$$CBLOF(x_i) = \begin{cases} |C_i| \times D(x_i, c_i) & \textit{if } C_i \in LC \\ |C_i| \times \min_{c_j \in LC} D(x_i, c_j) & \textit{if } C_i \in SC \end{cases} \tag{9}$$

Then, anomalies are characterized by a high CBLOF.

5.2 Experiments

Data. Due to the lack of readily available benchmarks for the problem outlined in Sec. 2, we experimentally evaluate on a real-world data set obtained from a large company. The provided data detail resource usage continuously tracked over a period of approximately two years. Since the usage is highly dependent on the time of day, we can generate 24 (hourly) data sets by grouping the usage data by hour. Each data set contains about 850 different time series. For a limited number of these series in each set we possess expert labels indicating either `normal` or `anomaly`.

Experimental setup. In turn, we treat each of the 24 data sets as the target domain and the remaining data sets as source domains. We consider transferring from a single source or multiple sources. Any labeled examples in the target domain are set aside and serve as the test set. First, the proposed decision functions are used to transfer instances from either a single source domain or multiple source domains combined to the target domain. Then, we train both the unsupervised CBLOF (Sec. 5.1), and supervised 1NN-DTW anomaly detection model that uses the labeled instances transferred to the target domain

(Sec. 4.4). Finally, both models predict the labels of the test set, and we report classification accuracy. For the density-based approach, we set the threshold on the final weights to 0.5. For the cluster-based approach we selected $\alpha = 0.95$, $\beta = 4$, and the number of clusters 10.

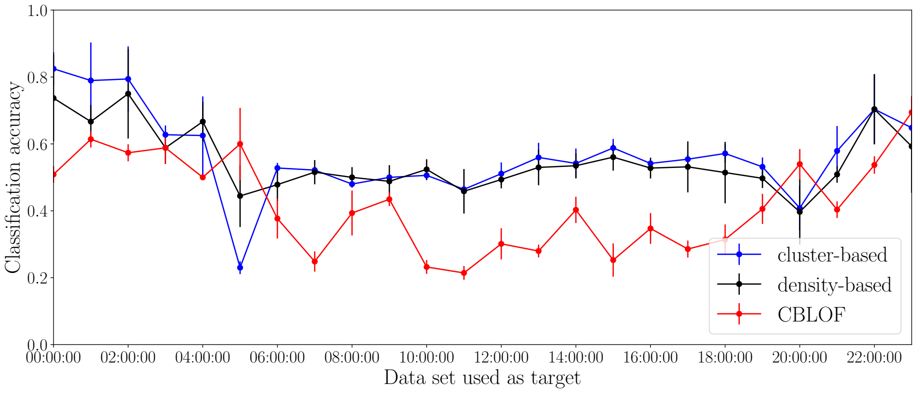

Fig. 1: The graph plots the mean classification accuracy and the standard deviation for each of the 24 (hourly) data sets. These statistics are calculated after considering 7 randomly chosen data sets as source domains, and performing the analysis for each combination of source and target. The plot indicates both transfer approaches with 1NN-DTW perform quite similarly, while outperforming the unsupervised method in 21 of the 24 data sets.

Evaluation. A limited excerpt of the experimental results is reported in Table 1. Figure 1 plots the full experimental results in a condensed manner. From the results we derive the following observations. First, instance transfer with 1NN-DTW outperforms the unsupervised CBLOF algorithm in 21 of the 24 data sets. Clearly, this indicates that the instances that are transferred by both decision functions, are useful in detecting anomalies. Second, the transfer works both between similar and dissimilar domains. To see this, one must know that in our real-world data set resource usage during the night is very different from usage during the day. As a result, the data sets at 00:00 and 01:00 are fairly similar for example, while data sets at 21:00 and 15:00 are highly different. From Table 1 it is clear that this distinction has little impact on the performance of the 1NN-DTW model. Third, the cluster-based decision function performs at least as well as the density-based variant. This is apparent from Figure 1.

6 Conclusion

In this paper we introduced two decision functions to guide instance-based transfer learning in case the instances are time series and the task at hand is anomaly detection. Both functions are based on two commonly knowns insights about anomalies: they are infrequent and unexpected. We experimentally evaluated

Table 1: A limited excerpt of the experimental evaluation. The number of transferred instances is denoted by n_A. *Density-based* is the density-based decision function with 1NN-DTW anomaly detection. *Cluster-based* is the cluster-based decision function with 1NN-DTW. *CBLOF* is the unsupervised anomaly detection. All reported numbers are classification accuracies on a hold-out test set in the target domain, rounded off. *Combo* is the the combination of 7 separate, randomly chosen source domains.

		Cluster-based		Density-based		CBLOF
Source	Target	n_A	Result	n_A	Result	Result
01:00	00:00	14	**89%**	13	**89%**	58%
03:00	00:00	11	**79%**	11	74%	52%
21:00	00:00	10	**79%**	9	58%	52%
combo	00:00	60	**90%**	46	85%	63%
03:00	06:00	6	**52%**	5	**52%**	39%
11:00	06:00	15	**56%**	8	**56%**	35%
21:00	06:00	7	**52%**	8	48%	39%
combo	06:00	79	**57%**	54	44%	35%
03:00	15:00	6	**58%**	5	**58%**	23%
11:00	15:00	19	**65%**	9	58%	30%
21:00	15:00	7	**58%**	7	**58%**	19%
combo	15:00	85	**67%**	54	54%	27%
03:00	19:00	6	**52%**	5	**52%**	44%
11:00	19:00	16	**60%**	8	48%	40%
21:00	19:00	7	**52%**	8	44%	40%
combo	19:00	81	**56%**	50	48%	44%

the proposed decision functions in combination with a 1NN-DTW classifier by comparing it to an unsupervised anomaly detection algorithm on a real-world data set. The experiments showed that the transfer-based approach outperforms the unsupervised approach in 21 of the 24 data sets. Additionally, both decision functions lead to similar results.

References

1. Andrews, J.T., Tanay, T., Morton, E., Griffin, L.: Transfer representation-learning for anomaly detection. ICML (2016)
2. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM computing surveys (CSUR) 41(3), 1–72 (2009)
3. Chattopadhyay, R., Sun, Q., Fan, W., Davidson, I., Panchanathan, S., Ye, J.: Multisource domain adaptation and its application to early detection of fatigue. ACM Transactions on Knowledge Discovery from Data (TKDD) 6(4), 18 (2012)
4. Fukunaga, K.: Introduction to statistical pattern recognition. Academic press (2013)
5. Kha, N.H., Anh, D.T.: From cluster-based outlier detection to time series discord discovery. In: Revised Selected Papers of the PAKDD 2015 Workshops on Trends and Applications in Knowledge Discovery and Data Mining-Volume 9441. pp. 16–28. Springer-Verlag New York, Inc. (2015)

6. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on knowledge and data engineering 22(10), 1345–1359 (2010)
7. Spiegel, S.: Transfer learning for time series classification in dissimilarity spaces. In: Proceedings of AALTD 2016: Second ECML/PKDD International Workshop on Advanced Analytics and Learning on Temporal Data. p. 78 (2016)
8. Torrey, L., Shavlik, J.: Transfer learning. Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques 1, 242 (2009)
9. Wei, L., Keogh, E.: Semi-supervised time series classification. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 748–753. ACM (2006)
10. Weiss, K., Khoshgoftaar, T.M., Wang, D.: A survey of transfer learning. Journal of Big Data 3(1), 9 (2016)