Journal of Universal Computer Science, vol. 25, no. 1 (2019), 2-15
submitted: 25/1/18, accepted: 28/10/18, appeared: 28/1/19 © J.UCS

Balanced Efficient Lifelong Learning (B-ELLA) for Cyber Attack Detection

Rafał Kozik
(UTP University of Science and Technology
Bydgoszcz, Poland
rkozik@utp.edu.pl)

Michał Choraś
(UTP University of Science and Technology
Bydgoszcz, Poland
FernUniversität in Hagen, Germany
chorasm@utp.edu.pl)

Jörg Keller
(FernUniversität in Hagen, Germany
joerg.keller@fernuni-hagen.de)

Abstract: This paper outlines and proposes a new approach to cyber attack detection on the basis of the practical application of the efficient lifelong learning cybersecurity system. One of the main difficulties in machine learning is to build intelligent systems that are capable of learning sequential tasks and then to transfer knowledge from a previously learnt foundation to learn new tasks. Such capability is termed as Lifelong Machine Learning (LML) or as Lifelong Learning Intelligent Systems (LLIS). This kind of solution would promptly address the current problems in the cybersecurity domain, where each new cyber attack can be considered as a new task. Our approach is an extension of the Efficient Lifelong Learning (ELLA) framework. Hereby, we propose the new B-ELLA (Balanced ELLA) framework to detect cyber attacks and to counter the problem of network data imbalance. Our proposition is evaluated on a malware benchmark dataset and we achieve promising results.

Key Words: Lifelong Machine Learning, Classification, Data imbalance, Cybersecurity, Malware detection
Category: L.4.0

1 Introduction

Network and information security are now one of the most pressing problems of homeland security, as they affect the economy, citizens, and whole societies directly. It is universally observed that the number of successful attacks on information, civilians, even seemingly secure financial systems and most importantly critical infrastructures [Kozik et al., 2015] is still growing [Kozik et al., 2016, Choraś et al., 2016]. One of the reasons lies in the inefficiency of signature-based approaches to detect cyber attacks. In situations, where new attacks (or even slightly modified families of malware) emerge continuously, the standard protection systems are not adequate until the new signatures are created [Choraś et al., 2013]. On the other hand, anomaly-based approaches (systems which detect abnormalities in traffic, e.g. abnormal requests to databases) [Andrysiak et al., 2014, Saganowski et al., 2013] tend to produce false positives (false alarms).

Therefore, our objective is to address the demand of developing a system that does not need to return to the previously learnt knowledge or data (e.g. in previous generations or learning phases) since the knowledge would be already preserved, encoded and embedded in the trained components. Such a capability of intelligent systems (called lifelong machine learning) is currently vital in cybersecurity, where each new cyber attack type can be considered as a new task.

In our previous related papers, we have only presented the concept (without the initial results) of applying a lifelong learning intelligent system (LLIS) to cybersecurity [Choraś et al., 2017], and we have pondered the data imbalance problems [Kozik and Choraś, 2016]. It is our firm belief that lifelong machine learning systems can overcome the limitations of statistical learning algorithms which need immense numbers of training examples and are suitable for isolated single-task learning [Chen and Liu, 2016].

In the current work we propose, implement and evaluate the practical solution. We present the practical application of the lifelong learning approach to cyber attack detection and a practical solution to the data imbalance problem.

The major contribution of this paper is our extension of the ELLA framework [Ruvolo and Eaton, 2013] to detect cyber attacks and cope with the problem of data imbalance (hereby termed as B-ELLA). Moreover, we evaluated our solution, termed B-ELLA, and we report the promising results.

The original ELLA framework allows for building and maintaining a sparsely shared basis for task models (so-called base classifiers). In the context of a malware detection problem, as considered in this paper, this basis can be perceived as patterns of behaviour that build up more complex behavioural models. The sparsity encourages knowledge transfers, which means that certain patterns can be shared among tasks. More precisely the detection model is composed as a linear combination of sparse vectors maintained on the shared basis. Thanks to the ELLA framework these vectors are constantly updated and used to approximate the base classifiers parameters.

The remainder of the paper is organized as follows: in Section 2 the state of the art in lifelong learning intelligent systems is presented. Section 3 contains the description of the new proposed lifelong learning B-ELLA approach for cybersecurity, while in Section 4 the results obtained on malware datasets are presented and discussed. Section 4 also contains a short description of the known ELLA framework. Conclusions are provided afterwards.

2 Overview of Lifelong Learning Intelligent Systems (LLIS)

Originally, lifelong learning was established as a sequence of learning tasks that need to be solved using the knowledge previously acquired and stored in classifiers that have already learnt [Chen and Liu, 2015]. According to [Pentina and Lampert, 2015] and [Pentina and Lampert, 2014], theoretical considerations on lifelong learning are relatively widely described in the literature, in particular in the light of the growing popularity of machine learning approaches and applications. However, scientific communities usually put more attention to aspects of learning based on well-known knowledge domains and well-labeled training datasets, while approaches to lifelong learning (or learning to learn) without observed data, e.g. to perform new, unforeseen tasks are not yet very popular. In [Baxter, 2000], one of the first attempts to describe the model of lifelong learning can be found. The author introduced a formal model called inductive bias learning, that can be applied when the learner is able to distinguish novel tasks drawn from multiple, related tasks from the same environment. Those considerations focused only on the finite-dimensional output spaces, and mainly on linear machines rather than nonlinear ones, in contrary to [Maurer, 2005], additionally extending earlier research with algorithmic stability aspects. In [Balcan et al., 2015], an approach to the problem of learning a number of different target functions over time is introduced, with assumptions that they are initially unknown for the learning system and that they share commonalities. Different approaches to solve this sequence of tasks include transfer learning [Segev et al., 2017], multitask learning, supervised, semi-supervised, reinforcement learning [Ammar et al., 2015], and unsupervised techniques. There are also works defining strong theoretical foundations for lifelong machine learning concepts. Particularly, in [Pentina and Lampert, 2014] authors worked on a PAC-Bayesian generalization bound applied for lifelong learning allowing quantification of relation between expected losses in future learning tasks and average losses in already observed (learnt) tasks. The bulk of approaches so far assume that the problem representation is not changing, (i.e. the feature space). It is a common method in classical event correlation based solutions [Choraś and Kozik, 2011, Choraś et al., 2011]. However, recent works increasingly consider that also the underlying feature space can fluctuate. To overcome those challenges, solutions such as changing kernels for feature extraction [Qiu and Sapiro, 2015], changing latent topics [Chen and Liu, 2014], or the underlying manifold in manifold learning [Yang and Crawford, 2016a, Yang and Crawford, 2016b] are proposed. The Hybrid Intelligent Systems [?] paradigm naturally addresses all the challenges of lifelong machine learning such as learning new tasks while preserving the knowledge of the preceding ones. In fact, classifier ensemble management resembles some of the algorithms proposed for lifelong learning. For example, a critical aspect of the lifelong learning systems is the ability to detect the task shift, which is quite similar to concept drift detection [Widmer and Kubat, 1996], and can be tackled by hyper-heuristics [Sim et al., 2015]. To deal with debatable cases in ensemble learning and to increase transparency in such debatable decisions, our hypothesis is that argumentation could be more effective than current resolution methods. Moreover, recent work on hybrid classifiers has demonstrated promising results of using an argumentation-based conflict resolution instead of voting-based methods for debatable cases in ensemble learning [Conţiu and Groza, 2016], showing that the hybridization of ensemble learning and argumentation fits the decision patterns of human agents.

In the next section, we propose the practical application of a lifelong learning framework to solve cybersecurity problems such as intrusion, anomalies, and cyber attack detection.

3 B-ELLA (Balanced ELLA) - the practical application of lifelong learning to cybersecurity

The concept of a task appears in many formal definitions of lifelong machine learning models [Pentina and Lampert, 2015]. For example, when considering telecommunication network monitoring for cyber security purposes, it is often difficult to distinguish when a particular task finishes and the subsequent one starts, i.e. when a different family of attacks has started. Therefore, the lifelong learning approach fits very well with the reality in the cybersecurity domain.

In practice, while designing and developing intelligent systems for anomaly and cyber threats detection one can draw the following conclusions:

- When it comes to the cybersecurity and cyber-attacks detection, there is no single classifier or IDS system that will allow the recognition of all kinds of attacks. Likewise, the same system (even if it learnt to detect the same type of attacks) has to be learnt again when changing the monitored network (topology, services, characteristics etc.). In that regards, we will need a transfer learning mechanism that will allow us to learn to detect attack B from knowledge acquired for attack A.

- There is an overlap of knowledge that an intelligent and adaptive system will need to be aware of. One can leverage this both to facilitate learning of new tasks and improving the effectiveness when executing the old ones. Using the cybersecurity example again, an IDS learnt in one network will use already established knowledge to detect attacks in another new network in a more accurate way (than without the lifelong learning approach).

In the following subsections, we present our approach to feature extraction, we discuss the previously conclusively proved ELLA framework, and we present our extension called Balanced-ELLA (B-ELLA).

3.1 Feature Extraction

The data acquired by the system has the form of NetFlows. NetFlow is a standardised format for describing bi-directional communication and contains information such as IP source and destination address, destination port, and the amount of the exchanged bytes.

It must also be noted that we do not address the problem of a realistic testbed infrastructure. The reason is that we have decided to evaluate the effectiveness of the proposed algorithms using the standard benchmark CTU-13 [Garcia et al., 2014] dataset, and we have followed the experimental setup of its authors in order to methodologically compare our results.

The CTU-13 dataset contains different scenarios representing different infections and malware communication schemes with a command and control centre.

A single NetFlow usually does not provide enough evidence to decide if a particular machine is infected, or if a particular request has malicious symptoms. Therefore, it is quite common [Garcia et al., 2014][Garcıa, 2014] for NetFlows to be aggregated into so-called time windows so that more contextual data can be extracted and malicious behaviour recorded (e.g. port scanning, packet flooding effects, etc.). In such approaches, various statistics are extracted for each time window.

The comprehensive overview of the feature extraction pipeline is presented in Fig.1. In general, the proposed feature extraction method aggregates the NetFlows within each time window. For each time window, we group the NetFlows by the IP source address. For each group (containing Netflows with the same time window and IP source address) we calculate the following statistics:

- number of flows
- sum of transferred bytes
- average sum of bytes per NetFlow
- average communication time with each unique IP address
- number of unique destination IP addresses
- number of unique destination ports
- most frequently used protocol (e.g. TCP, UDP).

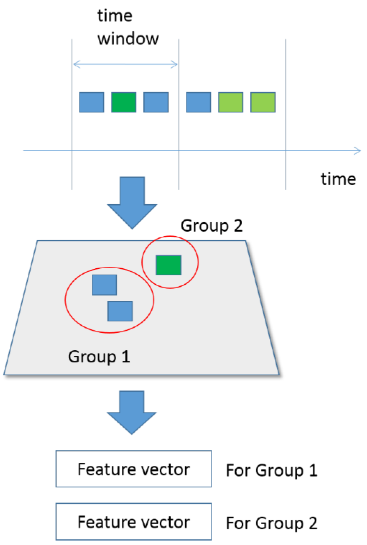

Figure 1: General overview of the feature vectors extraction pipeline.

3.2 ELLA Framework Overview

The ELLA framework [Ruvolo and Eaton, 2013] defines the lifelong learning problem as a series of supervised learning tasks, where each task $Z^{(t)} = \left(f^{(t)}, X^{(t)}, Y^{(t)}\right)$ is defined by training data $X^{(t)}$, training labels $Y^{(t)}$, and a prediction function (called in this paper also as a base learner) $f^{(t)} : X^{(t)} \rightarrow Y^{(t)}$.

The ELLA framework assumes that in each iteration the training algorithm may receive a batch of labelled data for some task t (a new one or previously trained). Moreover, it is assumed that the number of tasks is large and thus the ELLA algorithm must be scalable. As stated in [Ruvolo and Eaton, 2013], ELLA achieves equivalent accuracy to batch multi-task learning (MTL) [Ruder, 2017], has faster learning times and is able to train the model online.

ELLA framework uses a parametric model to represent the task-specific prediction function $f^{(t)}(x) = f(x, \theta^{(t)})$. The parameters θ are a linear combination of a so-called shared basis L, in the way that $\theta^{(t)} = Ls^{(t)}$.

The optimisation goal of the ELLA framework is to minimise the predictive loss over all training tasks. Formally the objective function is defined as:

$$e(L) = \frac{1}{T}\sum_{t=1}^{T} \min_{s^{(t)}} \left\{ \frac{1}{n_t}\sum_{i=1}^{n_t} \mathcal{L}\left(f(x_i^{(t)}; Ls^{(t)}), y_i^{(t)}\right) + \mu\|s^{(t)}\|_1 \right\} + \lambda\|L\|_2^2 \qquad (1)$$

where λ and μ are the regularization coefficients. However, to reduce the complexity related to the outer summation (over the number of training tasks T) the authors of the framework proposed to optimise $s^{(t)}$ only when training on task t. Moreover, the inner summation is approximated with the second order Taylor expansion around the optimal single task model. Therefore, the final objective function is defined as:

$$g(L) = \frac{1}{T}\sum_{t=1}^{T} \min_{s^{(t)}} \left\{ \frac{1}{n_t} \|\theta^{(t)} - Ls^{(t)}\|^2_{D^{(t)}} + \mu\|s^{(t)}\|_1 \right\} + \lambda\|L\|_2^2 \tag{2}$$

where $\theta^{(t)} = \min\limits_{\theta} \frac{1}{n_t} \sum_i^{n_t} \mathcal{L}\left(f(x_i^{(t)};\theta), y_i^{(t)}\right)$ is the optimal single task model and $D^{(t)}$ is the Hessian of the loss function evaluated at $\theta^{(t)}$. The optimisation process of the ELLA framework has the following steps:

1. Using the base learner, optimal model parameters $\theta^{(t)}$ are calculated for a task t.

2. Using the current basis L the model parameter vector $\theta^{(t)}$ is reconstructed in the way that $\theta^{(t)} = Ls^{(t)}$, where

$$s^{(t)} = \arg\min_{s^{(t)}} \left(\mu\|s^{(t)}\|_1 + \|\theta^{(t)} - Ls^{(t)}\|^2_{D^{(t)}}\right) \tag{3}$$

3. The updated matrix L_{new} is calculated using the $s^{(t)}$ from a previous step, solving the convex optimisation problem:

$$L_{new} = \arg\min_{L} \left(\lambda\|L\|_2^2 + \frac{1}{T}\sum_{1}^{T}(\|\theta^{(t)} - Ls^{(t)}\|^2_{D^{(t)}}\right) \tag{4}$$

From the optimisation point of view the formula (4) can be solved using LASSO (Least Absolute Shrinkage and Selection Operator) regression method. On the other hand, to find the optimal L one can first null the gradient and obtain the following formula:

$$\lambda L + \frac{1}{T}\sum_{1}^{T} D^{(t)}\left(\theta^{(t)} - Ls^{(t)}\right) {s^{(t)}}^T = 0 \tag{5}$$

It can be shown that this is a special case of linear matrix equation of form $AXB = C$,

$$B = \lambda I + \frac{1}{T}\sum_{1}^{T} D^{(t)} s^{(t)} {s^{(t)}}^T \tag{6}$$

and the constant C is equal to:

$$C = \frac{1}{T}\sum_{1}^{T} D^{(t)}\theta^{(t)}{s^{(t)}}^{T} \tag{7}$$

We can use the Kronecker product notation and the vectorisation (vec) operator to rewrite the equation. The vec operator indicates the vectorization of a matrix, which is a linear transformation converting the matrix into a column vector. Therefore, the equation can be rewritten as:

$$(B^{T} \otimes A)vec(x) = vec(C) \tag{8}$$

which has a closed-form solution in the form $H^{-1}b$, where

$$H = \lambda I + \frac{1}{T}\sum_{1}^{T}\left(s^{(t)}{s^{(t)}}^{T}\right) \otimes D^{(t)} \tag{9}$$

and

$$b = \frac{1}{T}\sum_{1}^{T} vec\left({s^{(t)}}^{T} \otimes \left(\theta^{(t)}D^{(t)}\right)\right) \tag{10}$$

The original ELLA framework considers two type of base learners algorithms, namely logistic regression and linear classifier. These have been chosen mainly due to the closed-form formula for Hessian matrix calculation.

3.3 Data Imbalance

The problem of data imbalance has recently been thoroughly studied [Kozik and Choraś, 2016, Wozniak, 2013] in the areas of machine learning and data mining. In many cases, this problem negatively impacts the machine learning algorithms and deteriorates the effectiveness of the classifier. Typically, classifiers in such cases will achieve higher predictive accuracy for the majority class, but poorer predictive accuracy for the minority class.

This phenomenon is caused by the fact that the classifier will tend to bias towards the majority class. Therefore, the challenge here is to retain the classification effectiveness even if the proportion of class labels is not equal. The imbalance of labels for the case of cyber security is significant. We may expect that only a few machines in the network will be infected and produce malicious traffic, while the majority will behave normally. In other words, most data contains clean traffic, while only a few data samples indicate malware.

The solutions for solving such a problem can be categorised as data-related and algorithm-related. The methods belonging to the data-related category use data over-sampling and under-sampling techniques, while the algorithm-related approaches introduce a modification to training procedures. This group can be further classified into categories using cost-sensitive classification (e.g. assigning a higher cost to majority class) or methods that use different performance metrics (e.g. Kappa metric).

In this paper we have used cost-sensitive learning as an effective solution for class-imbalance in large-scale settings. The procedure can be expressed with the following optimisation formula:

$$\hat{\theta} = \min_{\theta} \left\{ \frac{1}{2}||\theta||^2 + \frac{1}{2}\sum_{i=1}^{N} C_i||e_i||^2 \right\} \tag{11}$$

where θ indicates the classifier parameters, e_i the error in the classifier response for the i-th (out of N) data samples, and C_i the importance of the i-th data sample. In cost-sensitive learning, the idea is to give a higher importance to the minority class, so that the bias towards the majority class is reduced.

The original ELLA framework is designed to handle two types of machine learning algorithms, namely logistic regression or linear classifier. In this paper we have adapted the linear classifier. Therefore, the optimisation formula can be expressed using a matrix notation, as it is presented below:

$$\hat{\theta} = \min_{\theta} \left\{ (Y - X\theta)^T C(Y - X\theta) + \lambda||\theta||^2 \right\} \tag{12}$$

Setting the gradient to zero, it is easy to show that the closed form formula for finding the optimal θ is:

$$\hat{\theta} = \left(\frac{I}{\lambda} + X^T C X \right)^{-1} X^T C Y \tag{13}$$

Therefore, the Hessian matrix for a weighted linear classifier will also have a closed-form solution $X^T CX$.

4 Experiments

For the evaluation, we have used the CTU-13 dataset [Garcia et al., 2014] and the same experimental setup as its authors (to be able to compare our results). This dataset includes different scenarios which represent various types of attacks including several types of botnets. Each of these scenarios contain collected traffic in the form of NetFlows. The data were collected to create a realistic testbed. Each of the scenarios has been recorded in a separate file as a NetFlow using CSV notation. Each of the rows in a file has the following attributes (columns):

- StartTime - Start time of the recorded NetFlow,
- Dur - Duration,
- Proto - IP protocol (e.g. UTP, TCP),

- SrcAddr - Source address,
- Sport - Source port,
- Dir - Direction of the recorded communication,
- DstAddr - Destination Address,
- Dport - Destination Port,
- State - Protocol state,
- sTos - Source type of service,
- dTos - Destination type of service,
- TotPkts - Total number of packets that have been exchanged between source and destination,
- TotBytes - Total bytes exchanged,
- SrcBytes - Number of bytes sent by source,
- Label - label assigned to this NetFlow (e.g. Background, Normal, Botnet)

It must be noted that the ”Label” field is an additional attribute provided by the authors of the dataset. Normally, the NetFlow will have 14 attributes and the ”Label” will be assigned by the classifier.

Before using the B-ELLA framework to train the base classifiers, the raw NetFlows are processed in order to produce feature vectors. The procedure is detailed in Section 3.1. The procedure for calculating metrics is as follows:

1. NetFlows are separated into comparison time windows (we have used default time windows of 300s length).
2. Within the ground-truth NetFlow, labels are examined against the predicted ones and the tTP, tTN, tFP and tFN values (true and false positives and negatives) are amassed.
3. Recall, Precision, Accuracy, Error Rate and F-measure are estimated at the conclusion of each comparison time window.
4. Finally, when the whole file with NetFlows is processed, the final error metrics are calculated and produced.

Each time the algorithm spots a Botnet IP address in the comparison time window correctly, the True Positive counter value is raised. Likewise, a Normal IP address evaluated as a Not-Botnet address increments the True Negative

Table 1: Effectiveness of compared methods

Compared Methods	**Detection Ratio**	**False Positive Ratio**	**Accuracy**
Balanced ELLA	0.809 ± 0.093	0.017 ± 0.002	0.896 ± 0.046
Balanced STL	0.745 ± 0.104	0.034 ± 0.001	0.856 ± 0.052
Imbalanced ELLA	0.250 ± 0.121	0.000 ± 0.000	0.625 ± 0.061
Imbalanced STL	0.002 ± 0.018	0.000 ± 0.0001	0.501 ± 0.009

result. Each occurrence of a benign IP classified as a Botnet address increments the False Positive value. At every instance of a Botnet IP judged as Non-Botnet the False Negative counter is raised.

Each of the scenarios in the CTU dataset is considered as a separate task. For the evaluation purposes, we have generated 10 random splits of the dataset in order to produce training and test datasets.

The test environment we have used for experiments consisted of two machines equiped with 24GB of RAM and 8 CPU cores. To calculate features described in 3.1 we have used Apache Spark. The ELLA algorithm runs as a python program on one of these machines.

5 Results

The experimental results are presented in Tables 1 and 2. Table 1 contains the comparison of results obtained with the B-ELLA algorithm, balanced single task learner algorithm (STL), the original imbalanced ELLA algorithm, and imbalanced STL algorithm. The results show that balancing the base classifier embedded within the ELLA framework yields great improvements over the other evaluated methods.

In Table 2 we have presented the effectiveness of B-ELLA at a single task level. The average detection ratio is 80%, while the ratio of false positives (alarms) is less than 1%. As presented in Tables 1 and 2, the achieved results are very promising and motivate further work on both balanced lifelong learning systems (B-ELLA) and their application to cybersecurity.

6 Conclusions

In this paper, we have presented the new B-ELLA framework for cyber attack detection, where B-ELLA stands for Balanced Efficient Lifelong Learning. Our contributions are as follows: we proposed the extension of the conclusively proved ELLA framework to address the problem of data imbalance. Moreover, we presented an innovative practical implementation of the concept of lifelong learning

Table 2: Balanced ELLA: per-task effectiveness

Task	**Detection Ratio**	**False Positive Ratio**	**Accuracy**
1	0.400 ± 0.291	0.025 ± 0.003	0.688 ± 0.145
2	0.300 ± 0.085	0.027 ± 0.000	0.636 ± 0.042
3	0.800 ± 0.245	0.040 ± 0.002	0.880 ± 0.122
4	0.944 ± 0.045	0.015 ± 0.004	0.964 ± 0.022
5	0.960 ± 0.080	0.003 ± 0.002	0.978 ± 0.039
6	0.839 ± 0.072	0.001 ± 0.000	0.919 ± 0.036
7	1.000 ± 0.000	0.010 ± 0.001	0.995 ± 0.000
8	0.988 ± 0.025	0.007 ± 0.000	0.990 ± 0.013
9	0.862 ± 0.096	0.037 ± 0.001	0.912 ± 0.048
10	1.000 ± 0.000	0.008 ± 0.005	0.996 ± 0.003
Average	0.809 ± 0.093	0.008 ± 0.005	0.996 ± 0.003

to cyber attacks (malware) detection. We conducted experiments on a standard dataset with standard evaluation scenarios, and the achieved results demonstrate the efficiency of our approach.

References

[Ammar et al., 2015] Ammar, H. B., Tutunov, R., and Eaton, E. (2015). Safe policy search for lifelong reinforcement learning with sublinear regret. In *International Conference on Machine Learning*, pages 2361–2369.

[Andrysiak et al., 2014] Andrysiak, T., Saganowski, Ł., Choraś, M., and Kozik, R. (2014). Network traffic prediction and anomaly detection based on arfima model. In *International Joint Conference SOCO14-CISIS14-ICEUTE14*, pages 545–554. Springer.

[Balcan et al., 2015] Balcan, M.-F., Blum, A., and Vempala, S. (2015). Efficient representations for lifelong learning and autoencoding. In *Conference on Learning Theory*, pages 191–210.

[Baxter, 2000] Baxter, J. (2000). A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198.

[Chen and Liu, 2014] Chen, Z. and Liu, B. (2014). Topic modeling using topics from many domains, lifelong learning and big data. In *International Conference on Machine Learning*, pages 703–711.

[Chen and Liu, 2015] Chen, Z. and Liu, B. (2015). Lifelong machine learning in the big data era.

[Chen and Liu, 2016] Chen, Z. and Liu, B. (2016). Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 10(3):1–145.

[Choraś and Kozik, 2011] Choraś, M. and Kozik, R. (2011). Network event correlation and semantic reasoning for federated networks protection system. In *Computer Information Systems–Analysis and Technologies*, pages 48–54. Springer.

[Choraś et al., 2016] Choraś, M., Kozik, R., Flizikowski, A., Hołubowicz, W., and Renk, R. (2016). Cyber threats impacting critical infrastructures. In *Managing the Complexity of Critical Infrastructures*, pages 139–161. Springer.

[Choraś et al., 2011] Choraś, M., Kozik, R., Piotrowski, R., Brzostek, J., and Hołubowicz, W. (2011). Network events correlation for federated networks protection system. In *European Conference on a Service-Based Internet*, pages 100–111. Springer.

[Choraś et al., 2013] Choraś, M., Kozik, R., Puchalski, D., and Hołubowicz, W. (2013). Correlation approach for sql injection attacks detection. In *International Joint Conference CISIS12-ICEUTE´ 12-SOCO´ 12 Special Sessions*, pages 177–185. Springer.

[Choraś et al., 2017] Choraś, M., Kozik, R., Renk, R., and Hołubowicz, W. (2017). The concept of applying lifelong learning paradigm to cybersecurity. In *International Conference on Intelligent Computing*, pages 663–671. Springer.

[Conţiu and Groza, 2016] Conţiu, Ş. and Groza, A. (2016). Improving remote sensing crop classification by argumentation-based conflict resolution in ensemble learning. *Expert Systems with Applications*, 64:269–286.

[Garcıa, 2014] Garcıa, S. (2014). Identifying, modeling and detecting botnet behaviors in the network. *Unpublished doctoral dissertation, Universidad Nacional del Centro de la Provincia de Buenos Aires.*

[Garcia et al., 2014] Garcia, S., Grill, M., Stiborek, J., and Zunino, A. (2014). An empirical comparison of botnet detection methods. *computers & security*, 45:100–123.

[Kozik and Choraś, 2016] Kozik, R. and Choraś, M. (2016). Solution to data imbalance problem in application layer anomaly detection systems. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 441–450. Springer.

[Kozik et al., 2015] Kozik, R., Choraś, M., Flizikowski, A., Theocharidou, M., Rosato, V., and Rome, E. (2015). Advanced services for critical infrastructures protection. *Journal of Ambient Intelligence and Humanized Computing*, 6(6):783–795.

[Kozik et al., 2016] Kozik, R., Choraś, M., Renk, R., and Hołubowicz, W. (2016). Cyber security of the application layer of mission critical industrial systems. In *IFIP International Conference on Computer Information Systems and Industrial Management*, pages 342–351. Springer.

[Maurer, 2005] Maurer, A. (2005). Algorithmic stability and meta-learning. *Journal of Machine Learning Research*, 6(Jun):967–994.

[Pentina and Lampert, 2014] Pentina, A. and Lampert, C. (2014). A pac-bayesian bound for lifelong learning. In *International Conference on Machine Learning*, pages 991–999.

[Pentina and Lampert, 2015] Pentina, A. and Lampert, C. H. (2015). Lifelong learning with non-iid tasks. In *Advances in Neural Information Processing Systems*, pages 1540–1548.

[Qiu and Sapiro, 2015] Qiu, Q. and Sapiro, G. (2015). Learning transformations for clustering and classification. *The Journal of Machine Learning Research*, 16(1):187–225.

[Ruder, 2017] Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098.*

[Ruvolo and Eaton, 2013] Ruvolo, P. and Eaton, E. (2013). Ella: An efficient lifelong learning algorithm. In *International Conference on Machine Learning*, pages 507–515.

[Saganowski et al., 2013] Saganowski, Ł., Goncerzewicz, M., and Andrysiak, T. (2013). Anomaly detection preprocessor for snort ids system. In *Image Processing and Communications Challenges 4*, pages 225–232. Springer.

[Segev et al., 2017] Segev, N., Harel, M., Mannor, S., Crammer, K., and El-Yaniv, R. (2017). Learn on source, refine on target: a model transfer learning framework with random forests. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1811–1824.

[Sim et al., 2015] Sim, K., Hart, E., and Paechter, B. (2015). A lifelong learning hyperheuristic method for bin packing. *Evolutionary computation*, 23(1):37–67.

[Widmer and Kubat, 1996] Widmer, G. and Kubat, M. (1996). Learning in the presence of concept drift and hidden contexts. *Machine learning*, 23(1):69–101.

[Wozniak, 2013] Wozniak, M. (2013). *Hybrid classifiers: methods of data, knowledge, and classifier combination*, volume 519. Springer.

[Yang and Crawford, 2016a] Yang, H. L. and Crawford, M. M. (2016a). Domain adaptation with preservation of manifold geometry for hyperspectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(2):543–555.

[Yang and Crawford, 2016b] Yang, H. L. and Crawford, M. M. (2016b). Spectral and spatial proximity-based manifold alignment for multitemporal hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 54(1):51–64.