Online Influence Forest for Streaming Anomaly Detection

Inês Martins[1], João S. Resende[2], and João Gama[1]

[1] INESC TEC & Universidade do Porto
[2] NOVA LINCS & Universidade Nova de Lisboa
inesmartins@fc.up.pt jresende@fct.unl.pt jgama@fep.up.pt

Abstract. As the digital world grows, data is being collected at high speed on a continuous and real-time scale. Hence, the imposed imbalanced and evolving scenario that introduces learning from streaming data remains a challenge. As the research field is still open to consistent strategies that assess continuous and evolving data properties, this paper proposes an unsupervised, online, and incremental anomaly detection ensemble of influence trees that implement adaptive mechanisms to deal with inactive or saturated leaves. This proposal features the fourth standardized moment, also known as kurtosis, as the splitting criteria and the isolation score, Shannon's information content, and the influence function of an instance as the anomaly score. In addition to improving interpretability, this proposal is also evaluated on publicly available datasets, providing a detailed discussion of the results.

Keywords: Streaming data · Online · Incremental · Unsupervised · Anomaly detection · Ensemble · Kurtosis · Influence function

1 Introduction

The data revolution has branded the XXI century as the amount of data and heterogeneous platforms, responsible for mining information, constantly increase. Although this prospect provides meaningful patterns relevant in various fields such as healthcare and fraud detection, it also imposes privacy and security concerns, as well as efficient standardization to handle high speed and voluminous data, constantly expanding and evolving [1].

In anomaly detection, learning from data streams remains a challenge as it must consider an infinite and constantly changing nature that involves learning from imbalanced domains and forcing the evaluation process to encompass metrics that do not neglect the minority class [2]. Furthermore, the data flow depicted in most everyday scenarios matches the characteristics of a continuously evolving paradigm that introduces resource limitations and requirements for incremental and adaptive processing that delivers responses in a real-time fashion. As a result, concept drift, where the properties of the stream may change over time, is a major point of discussion. An effective mechanism for alleviating concept drift and improving the representation of under-represented values is

combining different base models in an ensemble approach. Ensemble methods for data stream mining have gained considerable popularity due to their high predictive capabilities, ability to confer robustness, and generalization [3].

Considering that anomalies are few and different compared to the rest of the data, the proposed method isolates anomalies, rather than profiling regular points, and attempts to determine the influence of each instance in the observed statistics in their groups. Unlike other methods that randomly select a splitting attribute, our approach favors the dimension that shows an increase in the fourth standardized moment (kurtosis), a measure of the heaviness of the tail of the distribution, as it is more likely to contain an outlier. This approach helps to tackle irrelevant dimensions that may lead to missing crucial anomalies [4].

In anomaly domains, it is relevant to identify abnormal or potentially defective events and localize the features that caused a distribution shift, which can be a critical step in the diagnosis. In this sense, our purpose is to design an online ensemble method that attempts to characterize the underlying distributions, isolating dynamics as they do not align with the expected behavior. Thus, the anomaly score is dictated by the complexity of the isolation process and the level of surprise given by the event's unpredictableness. Furthermore, as the influence of a sample that differs from the rest of the dataset tends to be larger than for normal points [5], the influence function of the proposed splitting heuristic will be used to score the deviation of an instance, measuring how deviating an example appears to be in a given distribution.

Therefore, the most significant contribution of this work is the design of a fully incremental and unsupervised anomaly detection strategy that focuses on identifying and curbing anomalous events by proposing online predictions where the algorithm responses are available sequentially over time. Moreover, to ensure a reliable representation of the evolving data characteristics that may lead to an obsolete model, this proposal also studies control mechanisms to examine the activity in the leaves and the consistency of the structure, that is, the ability to closely represent the observed behavior. Moreover, this procedure returns an anomaly score composed of three different metrics that could increase interpretability. Lastly, as it is imperative to attest to the effectiveness of the proposed methods in realistic scenarios, it also discusses and analyzes the results from testing this approach on publicly available real-time benchmark datasets with a distinctive number of points, dimensions, and anomalies.

Concisely, the paper is organized as follows: Section 2 provides a review of the current and most effective solutions that serve as motivation and inspiration to this work; Section 3 describes the implemented method from its principles to the basic unit of the ensemble and the anomaly score, completing with the pseudocode of an influence tree; Section 4 gathers the experimental trials and discussion of the results; finally, Section 5 closes this paper by stating final remarks and advancing future research directions and possible improvements.

2 Related work

With the volume and speed of real-time data increasing, obtaining large amount of labeled data, specially in an imbalanced scenario, is a topic of interest. In recent years, the attention to methods such as autoencoders [6] or random forest [7] have changed toward unsupervised approaches such as isolation forests [8], an ensemble method, Local Outlier Factor (LOF) [9] as a density-based clustering solution, or One-class SVM [10], a kernel-based unsupervised learning technique.

Since the increasing search for real-time and adaptive streaming solutions, the community dedicated their scope to improving and adapting batch solutions to a continuous processing setting. As an example, *Pokrajak et al.* [11] proposed an incremental version of LOF, where the the outlier factor is computed for each incoming data point, updating its statistics only with a few data points. Despite being an incremental method that can handle different densities and detect changes in data distributions, this solution demands high computational resources [12].

In real-time applications, predictions should be made online, where the algorithm identifies anomalies before incurring the actual event. Opposite to isolation trees, where both the split attribute and value are randomly selected to isolate abnormal instances at higher levels [8], *Putina et al.* [13] presents the Random Histogram Forest, an unsupervised and probabilistic approach, that builds a random forest based on the fourth central moment, also known as kurtosis, to guide the search for anomalous instances. In each leaf, the anomaly score, defined as the Shannon's information content, captures the likelihood of an example being an outlier [14]. Although it retains linear running time in the input size, this method does not implement an online streaming solution.

Although Isolation Forest [8] is an efficient method for anomaly detection with relatively low complexity, CPU, and time consumption, it requires all the data to build the forest, as well as pass over the dataset to assign an anomaly score. Thus, *Ding et al.* [15] adapted the isolation concept to streaming events using sliding windows. An important feature of this work is the ability to deal with concept drift by maintaining one input desired anomaly rate that determines if the detector is obsolete and if the latest data window should be used to build a new classifier.

Furthermore, *Tan et al.* [16] introduced a fast one-class anomaly detector for evolving data streams featuring an ensemble of random HS-Trees that does not require any data to build its structure. Unlikely Hoeffding Tree that induce decision trees and alter its structure dynamically by measuring the confidence of a splitting attribute heuristic as a new instance arrives [17], HS-Trees have a constant amortized time and memory complexity that records the mass profile of data operating with two consecutive windows where the learned profile is used to infer the anomaly scores of new data arriving in the latest window.

More recently, *Guha et al.* [4] proposed a non-parametric and unsupervised anomaly detection solution on streams based on the influence of an unseen point. This idea measures the externality imposed by that point by averaging the change in complexity. This ensemble of independent random-cut trees,

named Robust Random-Cut Forest (RRCF), provides a dynamically maintained strategy that allows incremental updates with as few changes as possible. Comparatively to other proposals, where node split is uniformly chosen at random, RRCF determines the dimension to cut proportionally to the attributes' range, which makes this solution more resilient to irrelevant dimensions.

3 Online and Incremental Influence Forest

Similarly to the isolation-based method introduced by *Liu et al.* [8], this work recursively splits the data through a tree. In the original proposal, anomalies are expected to be quickly isolated, lying closer to the root, whereas normal instances are located deeper. This proposal attempts to identify the feature that influences the distribution's shape by measuring the concentration of values around the mean and the tails. The statistical measure that accounts for both peakedness, the concentration of probability mass around the mean, and heavy-tailedness, extreme values occurring with nonnegligible probability, is given by the standard fourth moment coefficient of kurtosis. The kurtosis of a random variable X ($K[X]$) is defined in Equation 1, where μ and σ stand for the mean and standard deviation, respectively, and μ_4 represents the fourth central moment [18].

As it is perceived in Equation 1, the standardized data is raised to the fourth power, which implies that instances within the region of the peak have a negligible contribution to the kurtosis score, while extreme observations outside the region of the peak (e.g., outliers) contribute the most. Moreover, since kurtosis is a standardized measure that describes the shape of the distribution, it is invariant to scale or location.

$$K[\mathrm{X}] = E\left[\left(\frac{\mathrm{X}-\mu}{\sigma}\right)^4\right] = \frac{E[(\mathrm{X}-\mu)^4]}{E[(\mathrm{X}-\mu)^2]^2} = \frac{\mu_4}{\sigma^4} \tag{1}$$

Furthermore, influence functions are a classic technique from robust statistics that assesses how the model parameters change as a training point significance is increased by an infinitesimal amount [19]. Hence, this technique promotes the knowledge of the impact of data contamination when a point mass or perturbation is added to a statistic value to deviate it from the expected distribution.

The kurtosis influence function, $IF(x; K(.))$, described in Equation 2, which gives the name to this approach, provides a quantitative understanding of kurtosis ($K(.)$) when the contamination has occurred at point x. The expression, detailed by *Fiori et al.* [20], reveals that the contamination in both the tails and the center of the distribution increases this coefficient. Thus, as the influence function is unbounded, the kurtosis coefficient is sensitive to outlying values. Therefore, this formula estimates the contamination degree when an observation is added, helping to assess the impact of including a particular point and its degree of outlierness.

$$IF(x; K(.)) = \left(\left(\frac{x-\mu}{\sqrt{\mu_2}}\right)^2 - K(.)\right)^2 - K(.)(K(.)-1) - 4\frac{\mu_3}{\mu_2^{3/2}}\frac{x-\mu}{\sqrt{\mu_2}} \tag{2}$$

3.1 Influence Tree

Given a sample of data $X = x_1, ..., x_n$ of n instances from a d-variate distribution, to build a binary influence tree, the data space is recursively divided by selecting an attribute based on the heuristic measure, in this case, kurtosis, K. As this measure is expected to be affected by abnormal points, the highest value in this importance will indicate the presence of an outlier. However, it must be ensured that there is enough statistical evidence that the distribution has changed or that the number of instances processed is sufficient.

Similarly to the Hoeffding Trees [17], this approach wields Chebyshev's inequality, widely used in probability theory, to bound the tail probabilities of a random variable with finite variance. In particular, unlike other methods, this inequality can be applied to any distribution as long as it includes a defined variance and mean [21]. In other words, this will help to attest if, with a certain confidence, the heuristic measure has suffered an unexpected change.

Considering that X_a holds the highest observed K and, as depicted in Equation 3, if the last observation added forced the $K(A)$ to differ from its mean in more than t units, the probability is, at most, the quotient of the variance and the squared value of its distance to the mean. In other words, if the difference from its mean is significantly higher than some value t, the attribute with the highest importance shows enough evidence that an extreme value has been added. Thus, it can also be used as a splitting attribute of the node.

$$Pr[|X - E[X]| \geq t] \leq \frac{V[X]}{t^2} \tag{3}$$

Concerning the splitting criteria, this approach is more similar to the Hoeffding trees proposal for mining high-speed data streams, which essays to guarantee, with high probability, that the attribute with the highest heuristic is the best choice [17]. In addition, this proposal is also inspired by the Random Histogram Forest that uses kurtosis as its splitting heuristic [13]. Furthermore, when it comes to the splitting value, similarly to most of the state-of-the-art approaches, this study randomly chooses a value in the range of values determined so far.

Finally, the leaves update the sufficient statistics for each attribute when an instance is added to the sample. These statistics include the variables to assess kurtosis, influence function, range, and the sample size of observed data points, filtering the incoming instances according to the observed dynamic, and only expanding when there is enough evidence that the distribution has shifted. In particular, these numbers are constant, which means the complexity does not depend on the number of instances, only on the number of attributes.

3.2 Dynamic ensemble

One of the most common ensemble techniques is Bagging, which trains multiple base models with different points drawn by resampling the original dataset. In an online version, the forest trains several independent online influence trees to simulate the bootstrap process by sending a weight to each observation following a Poisson random variable [22]. This procedure adds another constant to define

the number of trees in the forest that run in parallel, given their independent nature.

The online influence forest proposed in this work is structured incrementally for streaming data that is supposed to be continuous and infinite. As a single instance is not sufficient to make inferences about a population distribution, the nodes define a minimum number of instances. However, as the stream of events progresses, each tree structure cannot grow indefinitely, constraining the tree depth to a maximum depth bound, user-configurable, to limit the height of each tree. Consequently, as predictions are made online, the algorithm response becomes available as the event is being processed, leading to lower scores and not flagging anomalies until enough points have been examined.

Moreover, attesting if the tree structures are consistent with the dynamics present in the available sample, that is, whether they are considered obsolete, is also decisive to ensure that the ensemble is capturing the new data properties and maintaining its integrity. With the limited size of each tree, an indication that the structure is becoming saturated and unable to adapt to new instances is the number of leaves that reaches the maximum height and shows evidence that a split must occur. Therefore, the number of saturated leaves is supervised to determine when each structure must be redefined.

This proposal implements additional reframe strategies to control the forest's accordance with the data, as illustrated in Figure 1. These strategies supervise the ability of the algorithm to reflect the current state of dynamics presented in the available data. Hence, the first strategy checks if, when a sample arrives at a leaf, the node is still active by checking the time between updates (Inactive Figure 1). A leaf is considered inactive when, on average, it has been enough time to record twice the minimum number of instances in a node. In this case, it might suggest that the parent node has picked the wrong split, and the splitting value is reframed. Lastly, another approach has been studied to tackle the change in dynamic or when the tree is considered saturated (Saturation Figure 1). This method maintains and reframes the tree structure from its root to the leaves by merging sibling nodes and replacing the parent node on higher levels. Thus, after the reevaluation, the number of leaves and levels reduce, and the original tree root, which holds the oldest distribution, is replaced.

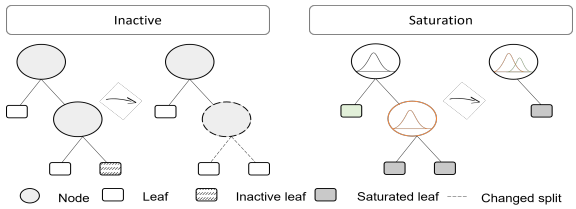


Fig. 1: Strategies to guarantee consonance between the tree structure and the observed data.

3.3 Anomaly score

The anomaly score vital in unsupervised methods is another crucial component that attempts to quantify the degree of discrepancy from the expected behavior according to a set of principles. It is also the only way to comprehend how a particular decision has been made. In most cases, measuring each observation and assessing why it has been given a degree of unexpectedness can provide more insights about the problem at hand than the predictive performance. The demand for higher explainability levels arises as the incompleteness of the problem formalization increases [23]. In particular, cybersecurity and fault tolerance are some domains that often require high levels of interpretability.

As this system was inspired by isolation forest [8] and random histogram forest [13], integrating the influence function, each observation will be described with an isolation score, the Shannon information content, and the expected value of the difference between the influence function and its average, for each attribute.

In this regard, the output of our framework consists of a tuple specifying three metrics. Firstly, as defined in Equation 4, the isolation score measures the average of the depth of each point from a collection of trees, $E[h(x)]$ and the average path length of an unsuccessful search in a binary search tree (BST). According to the expectation that anomalies will be filtered at higher levels, this formula returns a higher score for deviance values.

By defining a split based on the kurtosis statistic, when there is enough evidence that the distribution has changed, the leaves will become nodes, and the instances will not progress in the structure as the tree grows. The following scores will account for the density and the average poisoning when an observation is added to a leaf to survey the in-node distribution. Next, Equation 5 calculates the Shannon information content, level of surprise, by measuring the probability of the cardinality of the leaf over the number of seen examples. Hence, anomalies will record higher levels of Shannon's information. Finally, the influence function is added to the equation. As this estimator deems the effect of adding one point to the distribution, this function returns the degree of contamination that a specific instance implies to the leaf. Thus, this statistic is related to the anomalousness degree of observation in a particular distribution. As this work is designed for multivariate analysis, the influence score, shown in Equation 6, is given by the variability, over all attributes, of the kurtosis influence function when an example reaches a leaf ($IF(x; K(.))$).

$$c(n) = 2H(n-1) - (2(n-1)/n) \qquad isolation(x; n) = 2^{-\frac{E[h(x)])}{c(n)}} \tag{4}$$

$$P_{Leaf}[x] = \frac{|Leaf(x)|}{N} \qquad surprise(x) = \log\left(\frac{1}{P_{Leaf}[x]}\right) \tag{5}$$

$$influence(x) = E[(IF(x; K(.)) - E[IF(x; K(.))])^2] \tag{6}$$

Finally, the pseudo-code that illustrates the designed tree is represented in Algorithm 1.

Algorithm 1 kInfluence: Online and Incremental Influence Tree

Input : *node*: node of an influence tree;
Ex: Example of a Stream;
$K(.)$: Splitting evaluation heuristic;
δ: significance of choosing the correct splitting attribute;
N_{min}: minimum sample size to test splitting significance;
$maxDepth$: maximum depth a tree is allowed to grow.
Output: anomaly score metrics indexed by the row in stream
begin
  if *node is a leaf* **then**
    Update sufficient statistics (Subsection 3.1);
    Let $n \leftarrow$ sample size in leaf;
    if $n > N_{min}$ *and* Ex *is not empty* **then**
      if *node is inactive* **then**
        Reframe parent node (Figure 1);
      else
        Let X_a be the attribute with the highest $K(.)$;
        $p \leftarrow 1 - \frac{|K(A)-E[K(A)]|^2}{Var[K(A)]}$; (Equation 3)
        if $p < \delta$ **then**
          Let $h \leftarrow$ depth of the tree and $saturation \leftarrow \frac{\#saturated\_leaves}{\#leaves}$;
          if $h \geq maxDepth \wedge saturation > 0.5$ **then**
            Check for consistency and reframe tree (Figure 1);
          else
            Split Attribute $\leftarrow X_a$;
            Split Value $\leftarrow v \sim U(minX_a, maxX_a)$);
            Let $node.left \leftarrow$ left child and $node.right \leftarrow$ right child;
    Let $score \leftarrow \{index : [isolation, shannon, influence]\}$ (Subsecton 3.3)
  else
    $score \leftarrow kInfluence(node.left, Ex[node.X \leq node.v])$;
    $score \leftarrow kInfluence(node.right, Ex[node.X > node.v])$;
  return score;
end

4 Experimental Evaluation

Given the imbalanced nature of anomaly detection, the evaluation metric must be independent of the majority class. The precision, recall, and F1-score will be used in these experiments. While the recall is about completeness, concentrating on the percentage of correctly identified anomalies, precision calculates the rate of true positives over the detected anomalies, measuring the probability of correct detection of positive values and penalizing false alarms. Therefore, F_β is used to monitor several measures simultaneously. In this case, $\beta = 1$ assumes that precision and recall are equally important [2].

Hence, an experimental evaluation was conducted on open-source datasets from different domains to attest to the performance of the proposed method.

Table 1: Experimental trial metrics

Dataset				kInfluence		
Name	**#points**	**#dim.**	**%outliers**	Precision	Recall	F1
Ecoli[3]	336	7	2.6%	0.5	0.67	0.57
WBC[3]	278	30	5.6%	0.57	0.62	0.59
Ionosphere[3]	351	33	2.6%	0.30	0.55	0.39
Key Hold[4]	1883	1	0.006%	0.63	0.83	0.71
Key Updown[4]	5316	1	0.0008%	0.45	0.63	0.53
NYC[4]	10320	1	0.0005%	0.75	0.6	0.67

These examples, also considered in similar works, are available at Outlier Detection DataSets (ODDS) [24] and Numenta Anomaly Benchmark [25], a novel benchmark for evaluating online streaming anomaly detection applications. Table 1 summarizes the evaluation results where each row refers to a single dataset. The first four columns describe the data according to the number of instances, dimensionality, and the proportion of anomalies present. Besides the size difference in the first three datasets, these serve as multivariate analyses, and the last three as timeseries analyses. For these trials, as the algorithm parameters were kept constant, the procedures were conducted ten times to stabilize the outcomes, featuring a forest of 100 trees, 30 instances, and 95% confidence as the minimum number of points in the node and the probability to choose a split, respectively, and a maximum depth of 6. These values should be analyzed as a future direction, and each iteration will be plotted to understand this proposal's complexity and stabilizing times.

Although this proposal envisions an unsupervised learning method, an experimental evaluation, which should provide insights into how this approach behaves with distinct dimensions, sizes, or anomaly frequencies, compares the actual position of the outliers and the score information returned by our solution in a supervised manner. Given the online and incremental properties designed here, the algorithm requires a stabilizing time to accurately score points, as the first instances arriving will not be sufficient to make inferences. In this sense, to frame a realist scenario and not to compromise the performance, the outliers were randomly reorganized such that anomalies do not appear simultaneously or do not unfold in the first moments. As a result, only the timeseries datasets did not suffer any changes from the original form. Furthermore, as the similar approaches that inspired this work are designed with different characteristics or testing scenarios, their results will not be compared in this work. For the timeseries, despite working online and incrementally, our method missed one more anomaly than RRCF [4] and added one more point as critical, rendering a higher false alarm rate or lower recall.

[3] http://odds.cs.stonybrook.edu/[NAME]-dataset/

[4] https://github.com/numenta/NAB/tree/master/data/realKnownCause/

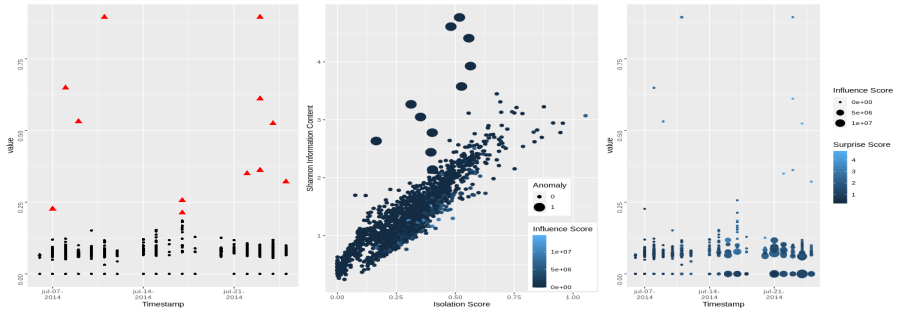
In a more detailed evaluation, as the last three datasets qualify as timeseries and facilitate the interpretation, they will be discussed closely, and the results and decision criteria will be analyzed.

The Key Hold dataset represents the timing of the key holds for several computer users, where the anomalies represent a change in the user. As critical anomalies are the ones that stand out amongst other points, it is possible to see different transitions reflecting an unexpected value for the key holds on that day. Figure 2a, from left to right, highlights the anomalies in red; the plot in the middle depicts the isolation, surprise, and influence score returned by our algorithm; and, finally, the last graph on the right investigates which observations classify with higher influence score as well as with higher surprise score. This figure shows that anomalies significantly differ from the rest of the values. Based on the definition of an anomaly, such points are more likely to appear on the upper side of the current trend. Furthermore, on the last graph, it is possible to see that the observations that score the highest ratings on the influence metric are usually points on the transition between values on similar timestamps, particularly after the first fortnight.

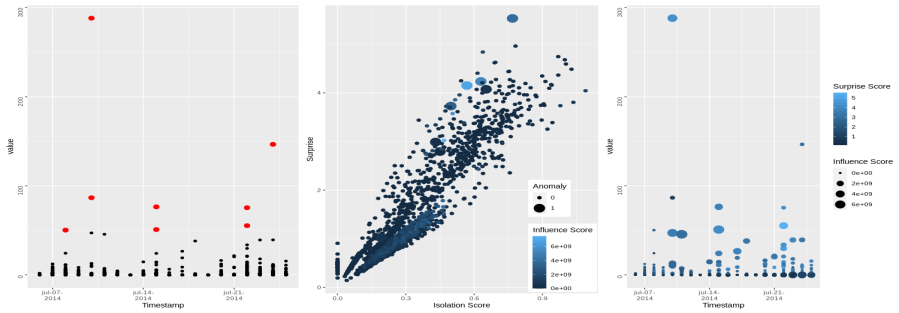
The Key Updown dataset describes keystrokes for several computer users, where the anomalies embody a change in the user. As assumed, abnormalities represent a significant transaction in their value. Figure 2b displays the anomalies in red; the metrics returned by our algorithm where, opposite from what is identified in the last trial, the influence score distinguishes points in the critical area. In particular, the last plot places outliers with influence, isolation, and surprise scores significantly more prominent than the surrounding observations. Moreover, as expected by the kurtosis and influence function, it is evident that orders that fall on the distribution's tails have their score increased, which is evident on the last graph where the tails of each timestamp are stressed as critical.

The NYC dataset corresponds to the number of NYC taxi passengers with five anomalies occurring during the NYC marathon, Thanksgiving, Christmas, New Year's day, and a snowstorm. The data file aggregates the total number of taxi passengers into 30-minute buckets. Therefore, to simplify, an anomaly often does not refer to a single observation but a time frame. Figure 2c illustrates the first timestamp with anomalous behavior issues. For instance, the first anomaly observed, the NYC marathon, has a lower value than the following numbers of passengers. From the metrics returned, depicted in the middle graph, there are not many anomalous points correctly predicted as critical. Since this dataset has many observations to be inspected with the naked eye, it is essential to examine the last plot to check which instances are flagged as dangerous. Thus, it is possible to see that the points with the most significant influence, isolation, and surprise scores have the highest number during the NYC marathon. Furthermore, despite their lower influence score, the isolation score spots the snowstorm as an outlier. The last anomaly correctly spotted was New Year's day, with a density and isolation score different from the peripheral. Another observation at the beginning of September registers similar scores as the New Year's day, which are

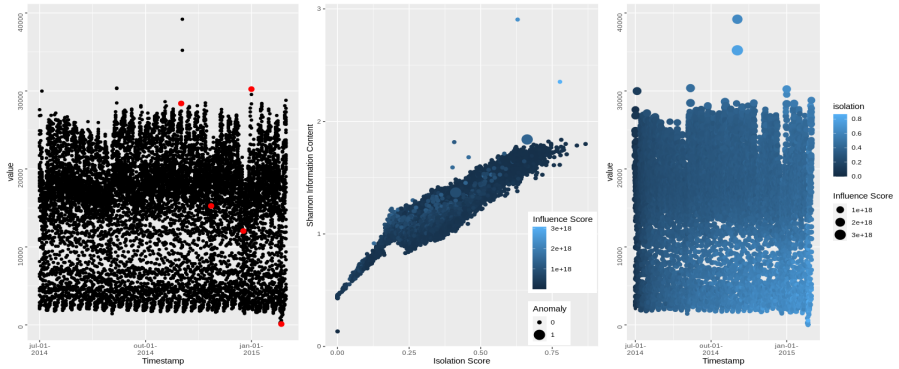
identical by analyzing the recorded value. However, our approach was not able to detect Thanksgiving and Christmas days.



(a) key Hold dataset



(b) Key Updown dataset



(c) NYC dataset

Fig. 2: Exprimental plots

5 Conclusion and Future Directions

This paper proposes an online, incremental, and unsupervised forest for streaming anomaly detection that focuses on selecting the best attribute according to the kurtosis score. Praising the interpretability of the output, the model definition of an anomaly captures both the complexity of isolating an outlier, the surprise level when an instance reaches a node, and the contamination effect imposed by a discrepant observation.

Given this proposal's online and incremental nature, this approach is essential in studying anomaly detection in streaming data. Despite implementing methods to avoid inadequate splits or obsolete structures, the next step will be to study the ability to adapt to dynamic changes, as well as further evaluate the effects of the required parameters, from tuning the number of necessary instances in the node to the number of trees or maximum height of the structure. Furthermore, a future step will be to study the repeatability and comparison with the approaches that inspired this work on a similar testing evaluation scenario.

Therefore, the next future direction should include parameter tuning and the benefit of maintaining a window with the latest points to control the consistency of the forest while evaluating the impact on the false alarm rate and recall to maximize the performance, bearing in mind an extensive comparison with the identical studies in the literature.

Acknowledgements This work has been supported by Fundação para a Ciência e Tecnologia (FCT), Portugal - `2021.04908.BD`, NOVA LINCS - `UIDB/04516/2020`, CityCatalyst - `POCI-01-0247-FEDER-046119`, financed by FEDER, and by the CHIST-ERA grant `CHIST-ERA-19-XAI-012`, and project `CHIST-ERA/0004/2019` and partially supported by the `CHIST-ERA` grant CHIST-ERA-19-XAI-012, funded by FCT. Also, this work is financed by the ERDF - European Regional Development Fund, through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme under the Portugal 2020 Partnership Agreement, within project City Analyser, with reference POCI-01-0247-FEDER-039924.

All the supports mentioned above are gratefully acknowledged.

References

1. S. R. et al., A survey on data preprocessing for data stream mining: Current status and future directions, Neurocomputing 239 (2017) 39–57.
2. P. Branco, L. Torgo, R. P. Ribeiro, A survey of predictive modeling on imbalanced domains, ACM computing surveys (CSUR) 49 (2) (2016) 1–50.
3. H. M. Gomes, J. Read, A. Bifet, J. P. Barddal, J. Gama, Machine learning for streaming data: state of the art, challenges, and opportunities, ACM SIGKDD Explorations Newsletter 21 (2) (2019) 6–22.
4. S. Guha, N. Mishra, G. Roy, O. Schrijvers, Robust random cut forest based anomaly detection on streams, in: International conference on machine learning, PMLR, 2016, pp. 2712–2721.

5. H. Thimonier, F. Popineau, A. Rimmel, B.-L. Doan, F. Daniel, Tracinad: Measuring influence for anomaly detection, arXiv preprint arXiv:2205.01362 (2022).
6. C. Zhou, R. C. Paffenroth, Anomaly detection with robust deep autoencoders, in: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, 2017, pp. 665–674.
7. L. Breiman, Random forests, Machine learning 45 (1) (2001) 5–32.
8. F. T. Liu, K. M. Ting, Z.-H. Zhou, Isolation forest, in: 2008 eighth ieee international conference on data mining, IEEE, 2008, pp. 413–422.
9. M. M. Breunig, H.-P. Kriegel, R. T. Ng, J. Sander, Lof: identifying density-based local outliers, in: Proceedings of the 2000 ACM SIGMOD international conference on Management of data, 2000, pp. 93–104.
10. B. a. a. Schölkopf, Support vector method for novelty detection, Advances in neural information processing systems 12 (1999).
11. D. Pokrajac, A. Lazarevic, L. J. Latecki, Incremental local outlier detection for data streams, in: 2007 IEEE symposium on computational intelligence and data mining, IEEE, 2007, pp. 504–515.
12. M. Salehi, L. Rashidi, A survey on anomaly detection in evolving data: [with application to forest fire risk prediction], ACM SIGKDD Explorations Newsletter 20 (1) (2018) 13–23.
13. A. Putina, M. Sozio, D. Rossi, J. M. Navarro, Random histogram forest for unsupervised anomaly detection, in: 2020 IEEE International Conference on Data Mining (ICDM), IEEE, 2020, pp. 1226–1231.
14. C. E. Shannon, A mathematical theory of communication, The Bell system technical journal 27 (3) (1948) 379–423.
15. Z. Ding, M. Fei, An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window, IFAC 46 (20) (2013) 12–17.
16. S. Tan, K. Ting, F. T. Liu, Fast anomaly detection for streaming data., in: Twenty-second international joint conference on artificial intelligence, 2011, pp. 1511–1516. doi:10.5591/978-1-57735-516-8/IJCAI11-254.
17. P. Domingos, G. Hulten, Mining high-speed data streams, Proceeding of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (11 2002). doi:10.1145/347090.347107.
18. N. Loperfido, Kurtosis-based projection pursuit for outlier detection in financial time series, The European Journal of Finance 26 (2-3) (2020) 142–164.
19. F. R. Hampel, The influence curve and its role in robust estimation, Journal of the american statistical association 69 (346) (1974) 383–393.
20. A. M. Fiori, M. Zenga, The meaning of kurtosis, the influence function and an early intuition by l. faleschini, Statistica 65 (2) (2005) 135–144.
21. M. Lovric, et al., International encyclopedia of statistical science, Springer Berlin Heidelberg, 2011.
22. N. C. Oza, S. J. Russell, Online bagging and boosting, in: International Workshop on Artificial Intelligence and Statistics, PMLR, 2001, pp. 229–236.
23. F. Doshi-Velez, B. Kim, Towards a rigorous science of interpretable machine learning, arXiv preprint arXiv:1702.08608 (2017).
24. S. Rayana, Odds library, http://odds.cs.stonybrook.edu/ (2016).
25. A. Lavin, S. Ahmad, Evaluating real-time anomaly detection algorithms–the numenta anomaly benchmark, in: IEEE ICMLA, 2015, pp. 38–44.