




# eif: Extended Isolation Forest

Sahand Hariri<sup>1</sup> and Matias Carrasco Kind<sup>2</sup>

<sup>1</sup> Mechanical Science and Engineering, University of Illinois at Urbana-Champaign, USA <sup>2</sup> National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign. 1205 W Clark St, Urbana, IL USA 61801

DOI: [00.00000/joss.00000](https://doi.org/00.00000/joss.00000)

## Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Submitted: 00 January 0000

Published: 00 January 0000

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

## Summary

The problem of anomaly detection has wide range of applications in various fields including scientific applications. anomalous data can have as much scientific value as normal data or in some cases even more. In this paper, we present an extension to the model-free anomaly detection algorithm, Isolation Forest. This extension, named Extended Isolation Forest (EIF), improves the consistency and reliability of the anomaly score produced by standard methods for a given data point. We show that the standard Isolation Forest produces inconsistent scores using score maps, and that these score maps suffer from an artifact produced as a result of how the criteria for branching operation of the binary tree is selected. We propose two different approaches for improving the reliability of anomaly detection. First we propose methods for transforming the data before the creation of each tree in the forest. Second, which is the preferred method of this paper, is to allow the slicing of the data to use hyperplanes with random slopes. This approach results in improved score maps. We show that the consistency and reliability of the algorithm is much improved using this extension by looking at the variance of scores of data points distributed along constant score lines. We find no appreciable difference in the rate of convergence nor in computational time between the standard Isolation Forest and EIF which highlights its potential as anomaly detection algorithm

## Motivation

While various techniques exist for approaching anomaly detection, Isolation Forest (Liu, Ting, and Zhou 2012) is one with unique capabilities. This algorithm can readily work on high dimensional data, it is model free, and it is computationally scalable. In the algorithm, data is sub-sampled, and processed in a tree structure based on random cuts in the values of randomly selected features in the data set. Those samples that travel deeper into the tree branches are less likely to be anomalous, while shorter branches are indicative of anomaly. As such, the aggregated lengths of the tree branches provide for a measure of anomaly or an “anomaly score” for every given point.

## The eif algorithm

---

**Algorithm 1**  $iForest(X, t, \psi)$

---

**Require:**  $X$  - input data,  $t$  - number of trees,  $h$  - sub-sampling size

**Ensure:** a set of  $t$   $iTrees$

1. **Initialize**  $Forest$

---

**Algorithm 1**  $iForest(X, t, \psi)$ 


---

2. set height limit  $l = \text{ceiling}(\log_2 \psi)$
  3. **for**  $i = 1$  to  $t$  **do**
  4.  $X' \leftarrow \text{sample}(X, \psi)$
  5.  $\text{Forest} \leftarrow \text{Forest} \cup iTree(X', 0, l)$
  6. **end for**
- 

---

**Algorithm 2**  $iTree(X, e, l)$ 


---

**Require:**  $X$  - input data,  $e$  - current tree height,  $l$  - height limit

**Ensure:** an  $iTree$

1. **if**  $e \geq l$  or  $|X| \leq 1$  **then**
  2.     **return**  $exNode\{Size \leftarrow |X|\}$
  3. **else**
  4. get a random normal vector  $\vec{n} \in \mathbb{R}^{|X|}$  where each coordinate is  $\sim \mathcal{N}(0, 1)$
  5. randomly select an intercept point  $\vec{p} \in \mathbb{R}^{|X|}$  in the range of  $X$
  6. set coordinates of  $\vec{n}$  to zero according to extension level
  7.  $X_l \leftarrow \text{filter}(X, (X - \vec{p}) \cdot \vec{n} \leq 0)$
  8.  $X_r \leftarrow \text{filter}(X, (X - \vec{p}) \cdot \vec{n} > 0)$
  9. **return**  $inNode\{$
  - $Left \leftarrow iTree(X_l, e + 1, l),$
  - $Right \leftarrow iTree(X_r, e + 1, l),$
  - $Normal \leftarrow \vec{n},$
  - $Intercept \leftarrow \vec{p}\}$
  10. **end if**
- 

---

**Algorithm 3**  $PathLength(x, T, e)$ 


---

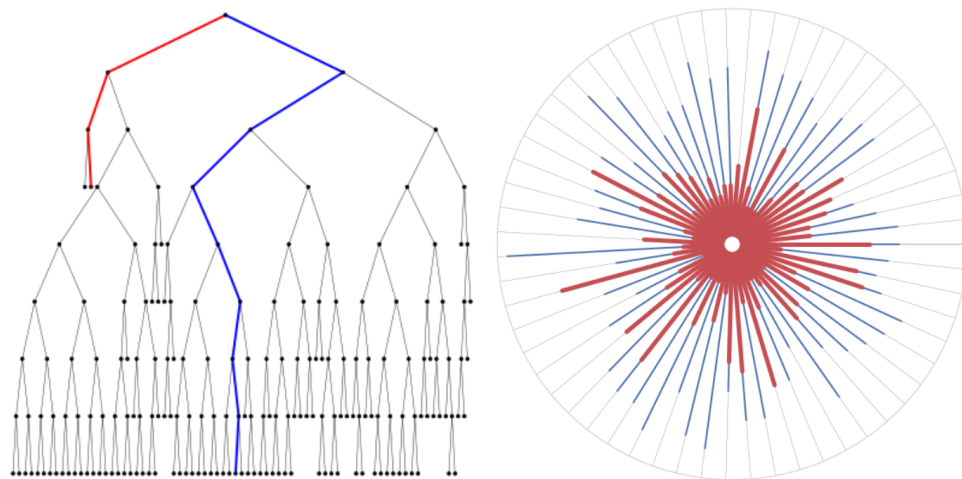
**Require:**  $\vec{x}$  - an instance,  $T$  - an  $iTree$ ,  $e$  - current path length; initialized to 0

**Ensure:** path length of  $\vec{x}$

1. **if**  $T$  is an external node **then**
  2.     **return**  $e + c(T.size)\{c(.)$  is defined in Equation 1}
  3. **end if**
  4.  $\vec{n} \leftarrow T.Normal$
  5.  $\vec{p} \leftarrow T.Intercept$
  6. **if**  $\{(\vec{x} - \vec{p}) \cdot \vec{n} \leq 0\}$  **then**
  7.     **return**  $PathLength(\vec{x}, T.left, e + 1)$
  8. **else if**  $\{(\vec{x} - \vec{p}) \cdot \vec{n} > 0\}$  **then**
  9.     **return**  $PathLength(\vec{x}, T.rigth, e + 1)$
  10. **end if**
- 

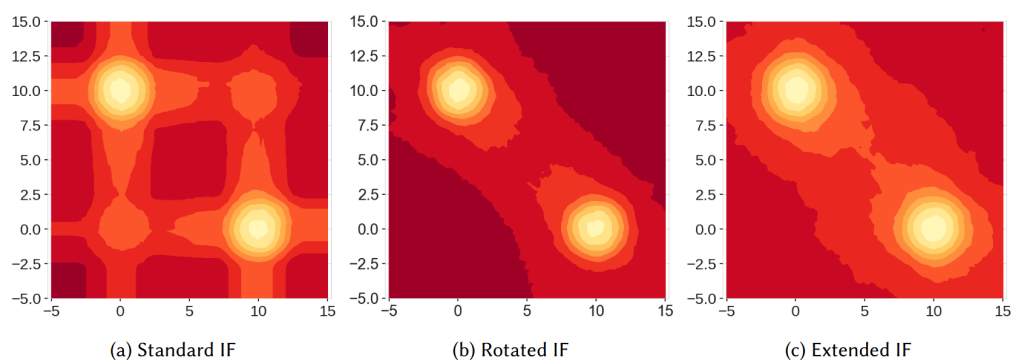
## Acknowledgements

MCK is supported by the National Science Foundation under Grant NSF AST 07-15036 and NSF AST 08-13543

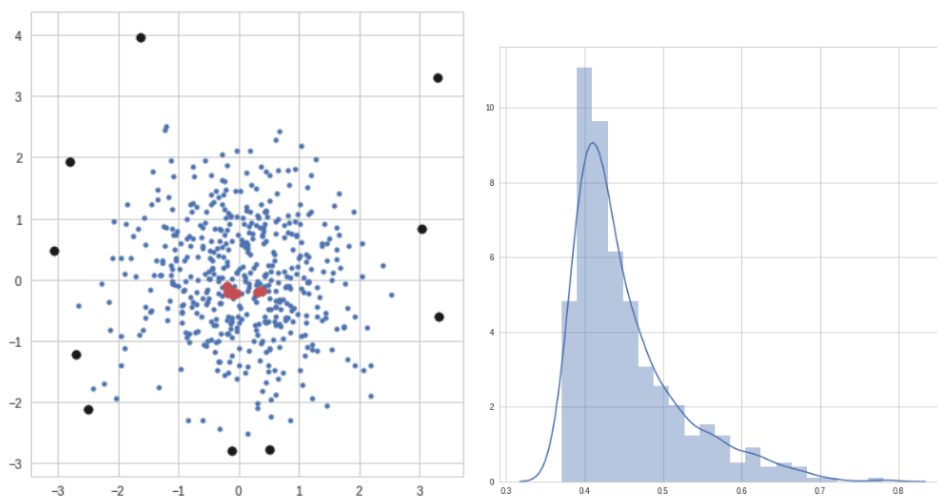


(a) Representation of a single tree in a forest. (b) Representation of a full forest where each radial line corresponds to a tree.

**Figure 1:** a) Shows an example tree formed from the example data while b) shows the forest generated where each tree is represented by a radial line from the center to the outer circle. Anomalous points (shown in red) are isolated very quickly, which means they reach shallower depths than nominal points (shown in blue).



**Figure 2:** Comparison of the standard Isolation Forest with rotated Isolation Forest, and Extended Isolation Forest for the case of two blobs.



**(a) Single 2D blob with anomalies (black) and nominal points (red).** **(b) Anomaly score Distribution**

**Figure 3:** a) Shows the dataset used, some sample anomalous data points discovered using the algorithm are highlighted in black. We also highlight some nominal points in red. In b), we have the distribution of anomaly scores obtained by the algorithm.

## References

Liu, Fei Tony, Kai Ming Ting, and Zhi-Hua Zhou. 2012. “Isolation-Based Anomaly Detection.” *ACM Trans. Knowl. Discov. Data* 6 (1). New York, NY, USA: ACM: 3:1–3:39. doi:[10.1145/2133360.2133363](https://doi.org/10.1145/2133360.2133363).