IEEE International Conference on Big Data 2019 – STREAM ML 2019

# Adaptive Hoeffding Tree with Transfer Learning for Streaming Synchrophasor Data Sets

**Zakaria El Mrabet, Daisy Flora Selvaraj, and Prakash Ranganathan**

[1]School of Electrical Engineering and Computer Science,
University of North Dakota

DECS

UND
UNIVERSITY OF NORTH DAKOTA

# Outline

- Introduction
- Problem Statement
- Objective
- Methodology
- Simulation Results
- Conclusion
- References

# Introduction

- the U.S. power grid reported a cyber-attack incident that disrupted grid operations, specifically transmission operators in the western power region are hit by a Denial of Service (DoS) attack, causing a temporary loss of visibility in certain sections of the SCADA system
- “the hack itself happened on March 5th, 2019, when a denial of service attack disabled Cisco’s adaptive security appliance devices, ringing power grid control systems in Utah, Wyoming, and California”. this is a major and the first cyber-attack on the U.S power grid [1].
- Other grid attacks include the Ukrainian power grid, which has experienced an outage for several hours impacting nearly 225,000 utility customers in 2015 due to cyber threats [2].
- One of the reasons behind these cyber-attacks and a number of major blackouts, especially in North America, is the lack of system awareness and reliable measurements [3]

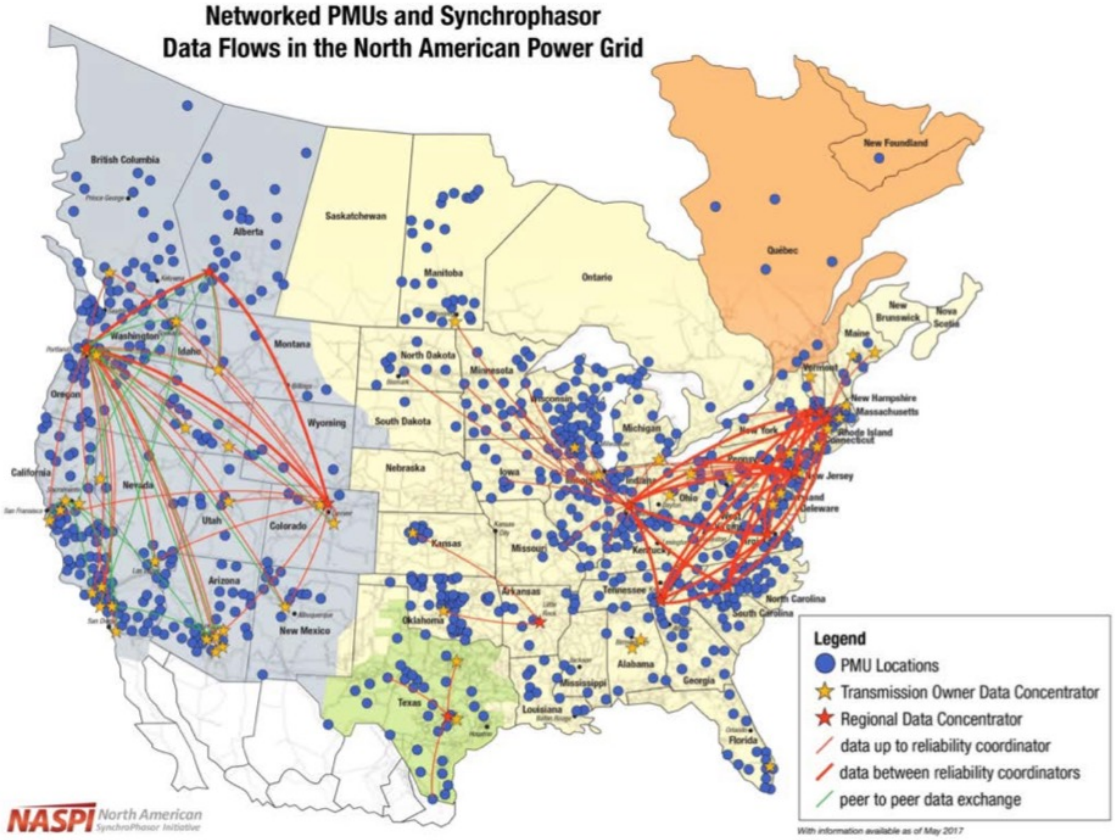
IMAGE

- Phasor Measurement Unit (PMU) constitutes a key sensor to improve the situational awareness and detect potential cyber physical attack

[1] “SECURITY: Experts assess damage after first cyberattack on U.S. grid.”
[2] D. Lee and D. Kundur, “Cyber attack detection in PMU measurements via the expectation-maximization algorithm,” in 2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2014, pp. 223–227.
[3] R. J. Campbell, “Electric Grid Cybersecurity,” p. 31.

# Problem Statement

- With more than 2500 PMUs deployed across the U.S. and Canada and each PMU generates 30/60 samples/s [1] (which represents 2592000/5184000 samples per day), extracting insight from this large amount of streaming data using the conventional machine learning approaches becomes challenging since they require loading the entire dataset for processing
- 1.5 TB/month streamed PMU data




Source [1]

[1] S. Silverstein, "2017 Synchrophasors and the grid. https://www.naspi.org/sites/default/files/reference_documents/naspi_naruc_silverstein_20170714.pdf.

# Problem statement

- The PMUs record the magnitude, phase angle, frequency, voltage and current phasors with a precise GPS based timestamp. This can significantly improve and facilitate the grid operations.
- A PMU can generate between 30-60 samples per seconds.
- the conventional machine learning cannot be applied to extract insights and detect potential anomalies (signatures) from this streaming PMU data. Additionally, these methods are not adaptable to concept drift events.
- The existing streaming machine learning approaches do not considered dealing with signatures with different durations.

# Objective

- Developing a streaming machine learning classifier called Transfer Adaptive Hoeffding tree (THAT) based on the Hoeffding tree and ADWIN including the transfer learning.
- Conduct a parametric study to define the appropriate configuration for the proposed model.
- Train and test the proposed model on four anomaly signatures, with different durations, including graduate concept drift events.
- Comparing the proposed approach against OzaBag based on several performance metrics including the accuracy, Kappa, and evaluation time.

# Methodology: PMU Dataset

- The dataset used in this study includes a collection of oscillatory events (e.g., four signatures) recorded by PMUs across multiple substations at various locations of a power system [1].
- Each signature is identified by its oscillation frequency, duration, and the potential cause. Additionally, a gradual concept drift event in introduced with each signature. This type of concept drift is selected since is mimic the most fault progression in the power system [2].
- The final dataset includes four signatures each of which includes 2000 normal events and 2000 oscillation events with gradual concept drifts.

| Signatures | Oscillation frequency | Duration | Potential event cause | Classes | Concept drift |
|---|---|---|---|---|---|
| Signature 1 | 0.1 Hz – 0.15 Hz | >> 400s | Generators | Oscillation event | Gradual |
| | 0.1 Hz – 0.15 Hz | << 400s | - | Normal event | |
| Signature 2 | 0.15 Hz – 1 Hz | >> 120s | Local plant control | Oscillation event | |
| | 0.15 Hz – 1 Hz | << 120s | - | Normal event | |
| Signature 3 | 1.0 Hz – 5.0 Hz | >> 60s | Inter-area oscillation | Oscillation event | |
| | 1.0 Hz – 5.0 Hz | << 60s | - | Normal event | |
| Signature 4 | 5.0 Hz – 14.0 Hz | >> 50s | Local plant control | Oscillation event | |
| | 5.0 Hz – 14.0 Hz | << 50s | - | Normal event | |

[1] "Test Cases Library." [Online]. Available: http://web.eecs.utk.edu/~kaisun/Oscillation/actualcases.html. [Accessed: 20-Sep-2019]
[2] U. Adhikari, T. H. Morris, and S. Pan, "Applying Hoeffding adaptive trees for real-time cyber-power event and intrusion classification," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 4049–4060, 2017.

# Methodology: THAT

- Part 1: Create a new Hoeffding adaptative tree (HAT) using the training set S.
- HAT uses a Hoeffding bound that relies on the minimum number of arriving samples to define a certain confidence threshold to build trees. Thus, it doest not require loading the entire data into memory. The only information required is the algorithm itself, which stores enough statistics on in its leaves, and enables the tree to grow, and classify samples in real-time [1][2].

**Algorithm 1**. Transfer learning Hoeffding Adaptive Tree (THAT)
Input: training set S, S', $HT_{source}$
Output: $HT_{target}$

1. If $HT_{source}$ is NULL
2. Let $HT_{target}$ be a tree with a single leaf (the root)
3. For all instances in $S$ do
4. Sort instances into leaf $l$ using $HT_{target}$
5. Update sufficient statistics in $l$
6. Increment $n_l$, the number of instances seen at $l$
7. If $n_l$ mod $n_{min} = 0$ and instances seen at $l$ not all of same class
8. Compute $\overline{G_l}(A_i)$ for each attribute
9. Let $A_a$ be attribute with highest $\overline{G_l}$
10. Let $A_b$ be attribute with second highest $\overline{G_l}$
11. Compute Hoeffding bound $\epsilon = \sqrt{\frac{R^2 \ln(\frac{1}{\delta})}{2n_l}}$
12. If $A_a \neq A_0$ and $\overline{(G_l}(A_a) - \overline{G_l}(A_b) > \epsilon$ or $\epsilon < \tau)$
13. Replace $l$ with an internal node that splits on $A_a$
14. For all branches of the split do
15. Add a new leaf with sufficient statistics
16. End for
17. End if
18. End if
19. End for
20. Return $HT_{target}$
21. Else

Fig. ?. THAT model (part 1)

[1] A. Bifet, R. Gavaldà, G. Holmes, and B. Pfahringer, "Machine learning for data streams: with practical examples in MOA," *MIT Press*, 2018.
[2] A. Bifet and R. Kirkby, "DATA STREAM MINING A Practical Approach," 2009

# Methodology: THAT

- Two differents approaches are investigated to compute the spliting cretirion $G$, which measures the average amount of purity that is gained in each subset of a split and indicates how well a given attribute separates the training examples according to their target classification. These approches are: Information gain and gini index.

**Algorithm 1**. Transfer learning Hoeffding Adaptive Tree (THAT)
Input: training set S, S', $HT_{source}$
Output: $HT_{target}$

1. If $HT_{source}$ is NULL
2. Let $HT_{target}$ be a tree with a single leaf (the root)
3. For all instances in $S$ do
4. Sort instances into leaf $l$ using $HT_{target}$
5. Update sufficient statistics in $l$
6. Increment $n_l$, the number of instances seen at $l$
7. If $n_l$ mod $n_{min} = 0$ and instances seen at $l$ not all of same class
8. Compute $\overline{G_l}(A_i)$ for each attribute
9. Let $A_a$ be attribute with highest $\overline{G_l}$
10. Let $A_b$ be attribute with second highest $\overline{G_l}$
11. Compute Hoeffding bound $\epsilon = \sqrt{\frac{R^2 \ln(\frac{1}{\delta})}{2n_l}}$
12. If $A_a \neq A_0$ and $(\overline{G_l}(A_a) - \overline{G_l}(A_b) > \epsilon$ or $\epsilon < \tau)$
13. Replace $l$ with an internal node that splits on $A_a$
14. For all branches of the split do
15. Add a new leaf with sufficient statistics
16. End for
17. End if
18. End if
19. End for
20. Return $HT_{target}$
21. Else

Fig. ?. THAT model (part 1)

# Methodology: THAT

- **Information Gain**: If the distribution of the two classes (e.g., oscillation event class, and normal event class) in the PMU stream contains the probabilities p1, and p2 of the classes, then the entropy of a given attribute $A$ in a training data set $S$ is calculated by:

$$Entropy\ (A) = \sum_{i=1}^{n} -p_i log_2 p_i \qquad (1)$$

- Here $n$ is the number of classes and it is equal to 2. The attribute, $A$, is one of the selected features which could be voltage (V), frequency (f), current (I), or the phase angle (φ). Then, the Information Gain is computed by:

$$Information\ Gain\ (S, A) = Entropy(A) - \sum_{k \in (A)} \frac{|S_k|}{|S|} Entropy(S_k) \qquad (2)$$

- Where $k$ is the value of the attribute $A$, and $S_k$ is a subset of $S$ w
$A = k$.

# Methodology: THAT

- The Gini index is given by:

$$Gini(A) = 1 - \sum_{i=1}^{n} {p_j}^2 \qquad (3)$$

- Here $n$ is the number of classes (e.g., n=2 in our case).

- The hoeffding bound is computed such that probability $1 - \delta$ corresponds to a confidence value of $\delta \in \{1,0\}$, where the true mean of a random variable of range $R$ will not differ from the estimated mean after $n$ independent observations by more than $\epsilon$ and it is given by:

$$\epsilon = \sqrt{\frac{R^2 \ln(\frac{1}{\delta})}{2n}} \qquad (4)$$

# Methodology: THAT

- In part 2, transfer learning (TL) is applied where the knowledge gained between two HT models trained on two different signatures are transferred. Additionally, The model is scalable to transfer knowledge beyond two HT models.
- There are two type of TL [1]:

```
21. Else
22. HTtarget = HTsource
23. Q ← all attributes of S' not in HTtarget
24. For each attribute A' dequeued from Q
25.   For each training instance I in S'
26.     Classify I' using the HTtarget
27.     If I ' is predicted correctly then
28.       Do nothing
29.     Else
30.       Replace HTtarget 's class node with a new node for
    attribute A'
31.       Add a new branch to node A', labeled with A's value in I'
32.       Add a new leaf node labeled with I''s target class label
33.     End if
34.   End for
35. End for
36. Return HTtarget
37. End if
```

Fig. ?. THAT model (part 2)

- **The inductive TL**: is used when the source and target tasks are different, but they share some common features.
- **The supervised TL:** The supervised type is used when $|S_T| >> |S_S|$ and aims to improve the task learning of domain $D_T$ given $S_T$

[1] N. Segev, "Transfer Learning Using Decision Forests," Institute of Technology Elul, Haifa, 2016.

# Methodology: THAT

- In our context, the THAT model will be trained on four signatures (oscillation and normal events) with different magnitude and durations: 400s, 120s, 60s, and 50s, respectively. Thus, a supervised transfer learning can be applied to transfer knowledge between different THAT models since

$|S_4| >> |S_3| >> |S_2| >> |S_1|$.

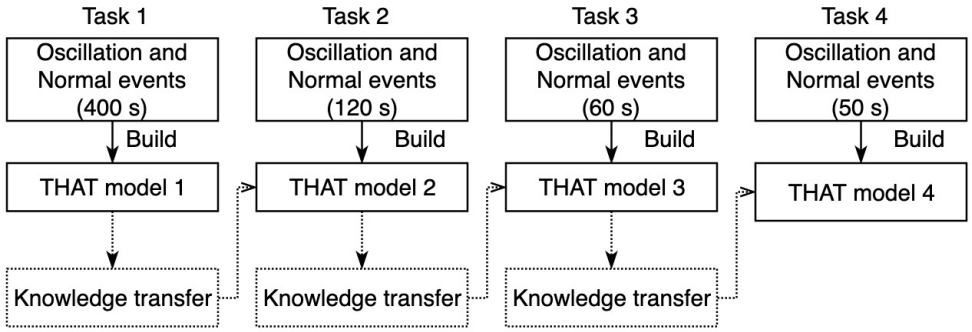



Fig. ?. Supervised transfer learning of THAT model trained on four oscillation events with different durations.

# Methodology: THAT

- To further improve the HAT model and make it adaptable to the eventual concept drift, ADWIN approach is included.
- ADWIN is an estimator with memory and change detector which is based on the sliding window approach for detecting changes through some statistical tests on different sub-windows.

**Algorithm 2**. ADWIN: Adaptive Windowing Algorithm [1]

1. Initialize Window $W$
2. for each $t > 0$
3. do $W \leftarrow W \cup \{x'_t\}$ (i.e., add $x'_t$ to the head of $W$)
4. repeat drop elements from the tail of $W$
5. until $|\hat{u}_{W0} - \hat{u}_{W1}| \geq \epsilon$ holds for every split of $W$ into $W = W_0 . W_1$
1. output $\hat{u}_W$

- The main advantage of ADWIN compared to other sliding window-based approaches is the adaptable window size. Instead of defining a fixed window size, ADWIN uses a dynamic window that adapts its size based on the rate of change of data within the window.

[1] Bifet and R. Gavalda, "Learning from time-changing data with adaptive windowing," in Proceedings of the 2007 SIAM international conference on data mining, 2007, pp. 443–448.

# Methodology: Performance metrics

- Accuracy:

$$Accuracy = \frac{TP + TN}{TP+TN+FN+FP} \quad (5)$$

- Kappa:

$$Kappa = \frac{(Accuracy - random\ accuracy\ )}{(1 - random\ accuracy)} \quad (6)$$

- Where $Accuracy$ is given by Equation (5), and $random\ accuracy$ is defined as:

$$random\ accuracy = \frac{(TN+FP)*(TN+FN)+(FN+TP)*(FP+TP)}{Total^2} \quad (7)$$

- Here $Total$ is $TP + TN + FN + FP$.

# Simulation Results

1. Experiment (I): THAT without supervised transfer learning

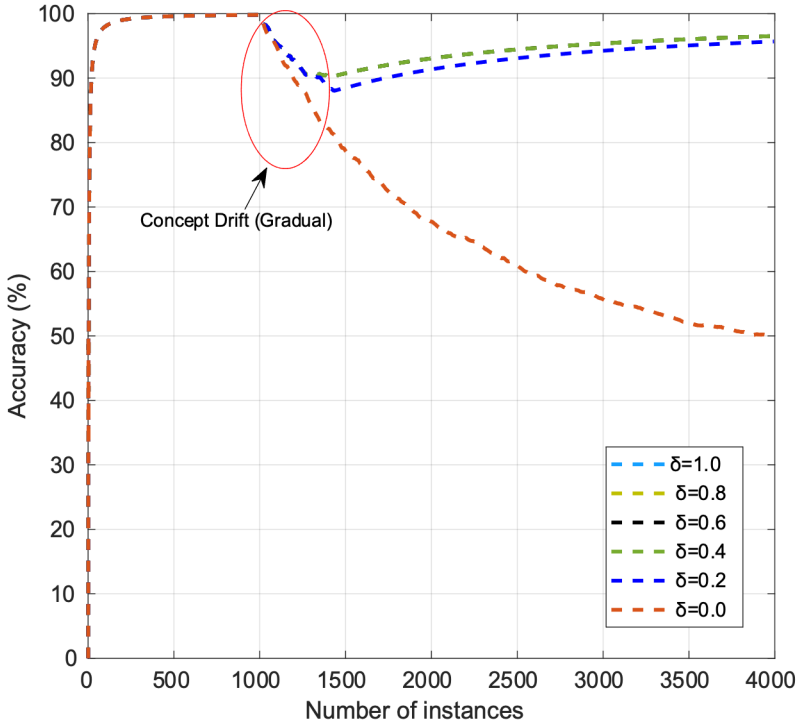



Fig. ?. Accuracy vs Number of Instances for THAT with Gini Index function and different $\delta$ values.

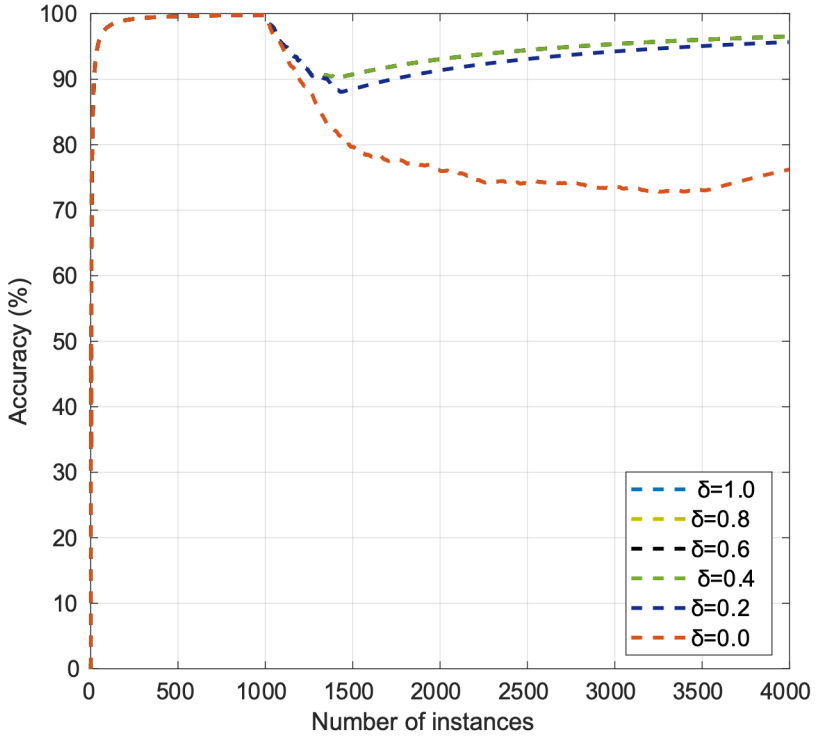



Fig. 3. Accuracy vs Number of Instances for THAT with Information Gain and different $\delta$ values.

# Simulation Results

1. Experiment (I): THAT without supervised transfer learning

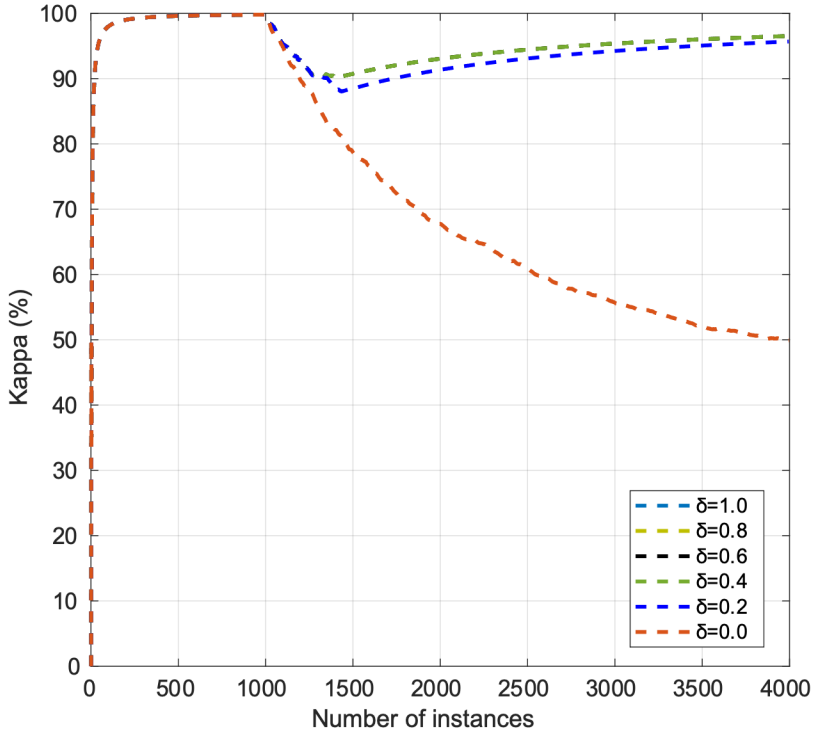



Fig. 4. Kappa vs Number of Instances for THAT with Gini Index function and different $\delta$ values.

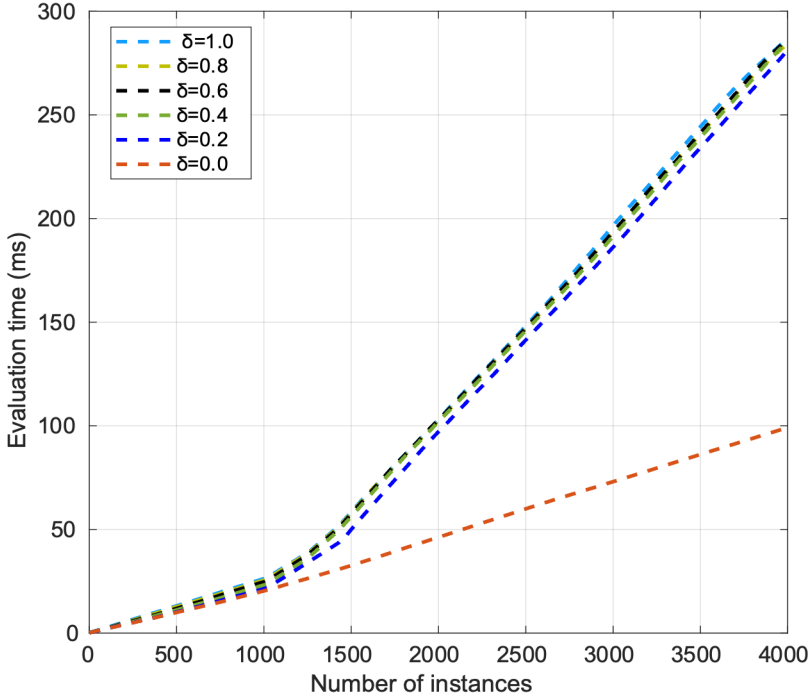



Fig. 5. Evaluation time vs Number of Instances for THAT with Gini Index function and different $\delta$ values.

# Simulation Results

1. Experiment (I): THAT without supervised transfer learning

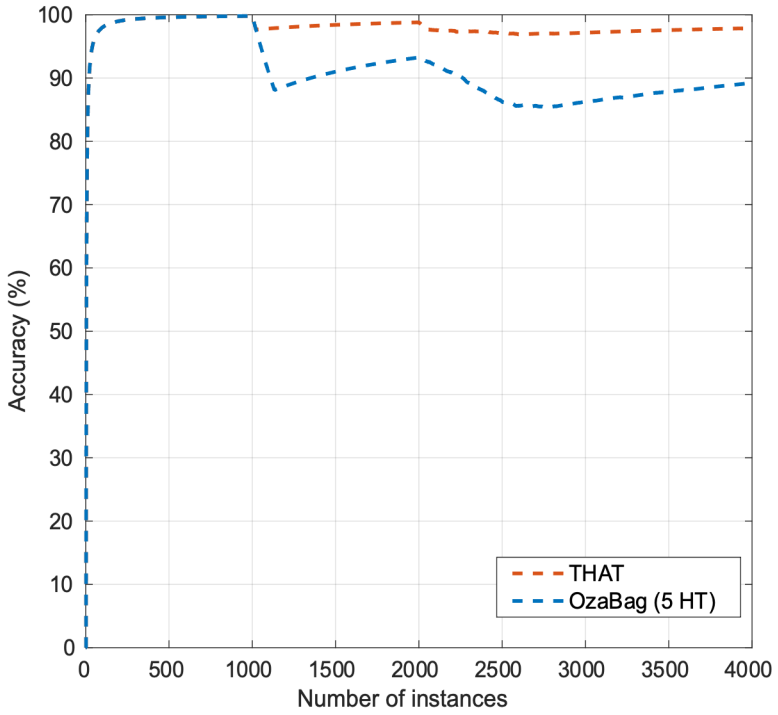



Fig. 4. THAT vs OzaBag in terms of accuracy as a function of the number of instances.

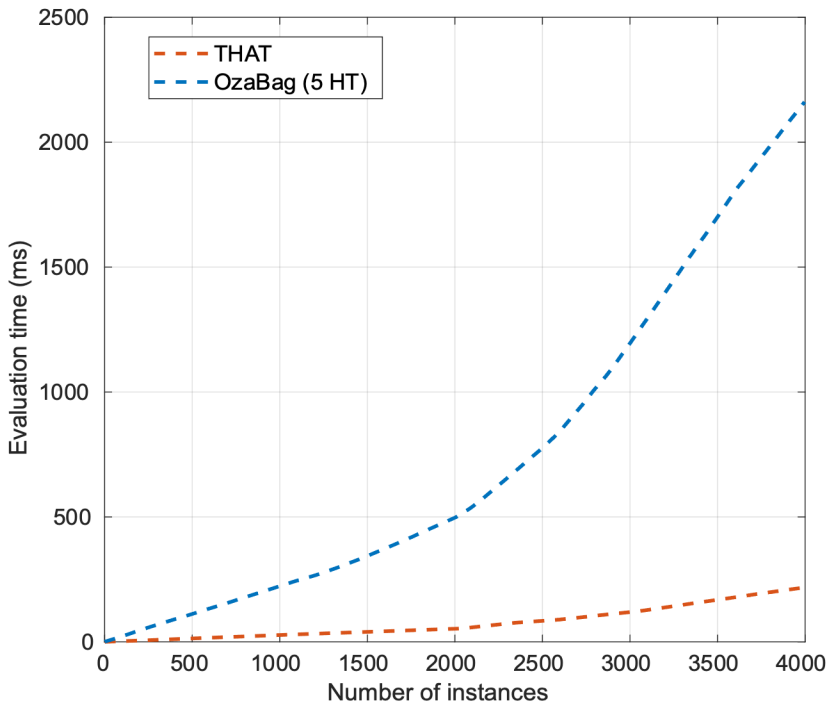



Fig. 5. THAT vs OzaBag in terms of evaluation time as a function of the number of instances.

# Simulation Results

1. Experiment (I): THAT without supervised transfer learning

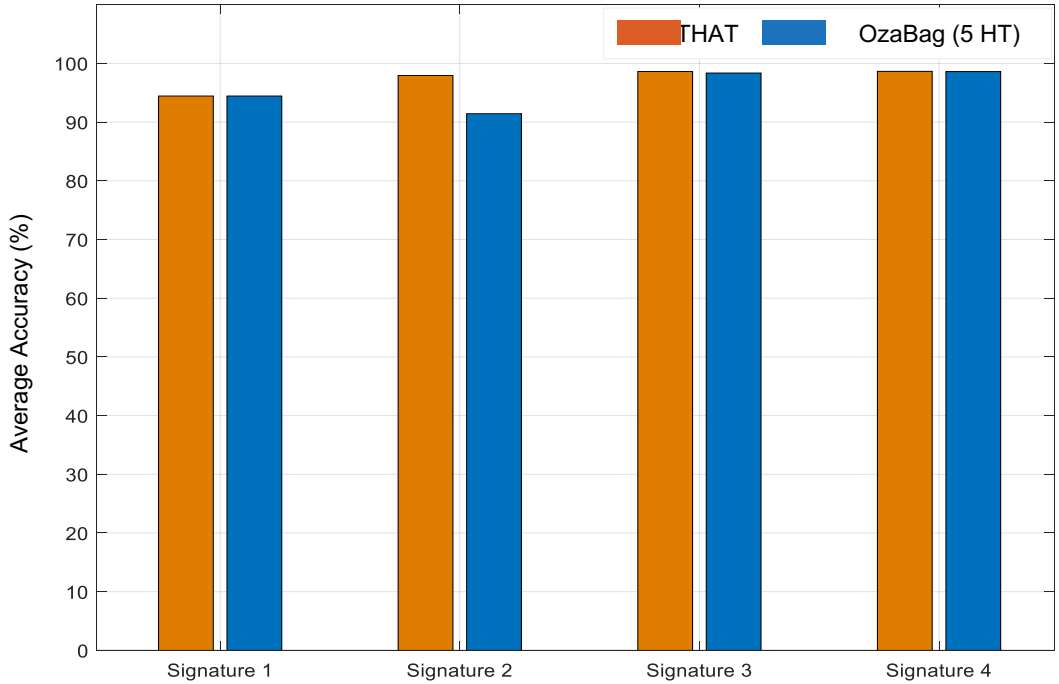



Fig. 8. THAT vs OzaBag in terms of average accuracy

# Simulation Results

1. Experiment (II): THAT with supervised transfer learning

TABLE 1. COMPARISON BETWEEN THAT MODEL AND OZABAG FOR 4 SIGNATURES (16,000 SAMPLES)

| Datastream models | Average accuracy | Evaluation time/instance |
|---|---|---|
| THAT model with supervised transfer learning | 94% | 0.34ms |
| OzaBag (5 HT) | 94% | 1.04ms |

# Conclusion

- A THAT model based on Hoeffding tree and ADWIN including transfer learning is developped to anomaly signature in PMU stream data.
- The THAT model are trained and tested on four signatures with different durations.
- Two set of experiments were considered: THAT with/without transfer learning.
- The THAT model is compared against OzaBag based on accuracy and evaluation time.
- In the first set, THAT and OzaBag models report higher average accuracy ranging between 91% and 99%. In the second one, the average accuracy of both models decrease to 94%, however, THAT model required smaller computational run-time than OzaBag.
- THAT model can be appropriate candidate to detect anomalies in PMU stream data in near-real time

# References

# Thank you!

# Questions?