Practical Application of Machine Learning based Online Intrusion Detection to Internet of Things Networks

Christopher Nixon
Staffordshire University
Stoke-on-Trent, UK
nr106584@student.staffs.ac.uk

Mohamed Sedky
Staffordshire University
Stoke-on-Trent, UK
M.H.Sedky@staffs.ac.uk

Mohamed Hassan
Staffordshire University
Stoke-on-Trent, UK
Mohamed.Hassan@staffs.ac.uk

***Abstract*—Internet of Things (IoT) devices participate in an open and distributed perception layer, with vulnerability to cyber attacks becoming a key concern for data privacy and service availability. The perception layer provides a unique challenge for intrusion detection where resources are constrained and networks are distributed. An additional challenge is that IoT networks are a continuous non-stationary data stream that, due to their variable nature, are likely to experience concept drift. This research aimed to review the practical applications of online machine learning methods for IoT network intrusion detection, to answer the question if a resource efficient architecture can be provided? An online learning architecture is introduced, with related IDS approaches reviewed and evaluated. Online learning provides a potential memory and time efficient architecture that can adapt to concept drift and perform anomaly detection, providing solutions for the resource constrained and distributed IoT perception layer. Future research should focus on addressing class imbalance in the data streams to ensure that minority attack classes are not missed.**

***Index Terms*—Internet of Things, Intrusion Detection, Online Machine Learning, Concept Drift, Anomaly Detection**

I. Introduction

The open and distributed architecture of Internet of Things (IoT) devices makes them vulnerable to both active and passive cyber attacks affecting the data privacy and availability of these services [1], [2]. Intrusion Detections Systems (IDS) are capable of both host and network based monitoring of cyber attacks [3]. IDS perform detection in several ways: *misuse detection* involves monitoring for known attack signatures; *anomaly detection* involves monitoring for deviations from known behaviour, enabling detection of unseen or zero day attacks; *hybrid detection* is a combination of misuse and anomaly detection [3]. Machine Learning (ML) techniques have been applied across all of the aforementioned detection methods [3]. IoT presents a unique challenge compared to traditional applications of IDS in so much that devices or sensors are constrained in performance capability and communicate over distributed, low power networking technologies, known as a *perception layer*, as described in [1] and [2]. Data gathered from the perception layer is transported to a back end *application layer* via a *transportation layer* consisting of traditional IP networking technologies [1], [2]. Machine Learning typically requires substantial computational resources to complete continuous testing and training of the model, this is a challenge within the perception layer where these resources are scarce [2], and requires consideration of IDS deployment architecture which, when *centralised* will become a single point of failure, with a *distributed* or *hybrid* architecture being preferred to ensure coverage [1], [2].

Communication at the perception layer and onwards via the transport layer, can be considered to be a continuous data stream which has no end [4]. ML methods map the posterior probability: $p(y|X)$, of a predicted class (y) given an observed feature (X). Given the variable nature of networks within the perception layer [1] and the possibility of new attacks, the data stream can be considered to be non-stationary whereby the posterior probability will change over time, degrading performance, this is known as real concept drift [5]. Signature based and offline trained IDS models cannot cope with concept drift, making necessary the use of online adaptive ML methods to ensure new concepts can be effectively learned and any loss in performance mitigated. A common challenge between IoT environments and online learning is resource constraints, where memory and processing time need to be effectively managed [1], [2], [5]. The aim of this research is to review the practical applications of online ML methods to IoT network intrusion detection, to answer the question of do online ML techniques provide a resource efficient architecture for IoT intrusion detection? The primary contribution of this research is the demonstration of practical online learning techniques to solve the resource constraints of the IoT perception layer for intrusion detection.

The remainder of this paper is organised as follows: section II, introduces the necessary architecture for online learning; section III, discusses online IDS approaches presented in related studies; section IV, provides evaluation results of informed and unsupervised approaches for IDS; section V, discusses how online learning can provide a resource efficient architecture for IoT intrusion detection; and section VI, presents conclusions.

II. Online Learning Architecture

Traditional IDS ML studies focus on offline batch learning [3], [6], whereby the model is trained and tested using all of the data set at once. Failure to update the model over time to adapt to concept drift will result in degradation of performance as demonstrated by [7]. Online learning necessitates that each data item can only be processed once, requiring incremental or interleaved-test-then-train evaluation, whereby a model is tested and then trained once on each individual unseen data item within the stream to facilitate loss estimation [5]. Evaluation of online algorithms requires measurement of performance loss, time complexity and memory consumption over time [5], [8].

An example online learning architecture is given in fig. 1. The components of which are briefly discussed below:

- *Forgetting Mechanism*: allows for management of the model's memory footprint by implementing a sliding window or fading factor so that only most relevant examples are remembered [5].
- *Classifier*: implements an interleaved test-then-train evaluation, estimating loss based on testing new observations [5].
- *Change Detector*: detects concept drift by either monitoring the change in distribution of a detection window or using statistical techniques to monitor changes in loss estimation, referred to as statistical process control (SPC) [5]. When change is detected, a warning or change signal is sent to the classifier based on predefined thresholds to signal training data collection and retraining [5], [9].

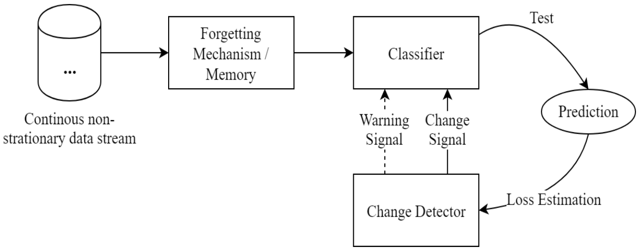

Fig. 1. Online Learning Architecture

III. Online IDS Approaches

Whilst broader studies have covered application of online techniques to varying use cases [8], there are fewer complete examples for IDS. In this section more recent studies are introduced and briefly evaluated covering the areas of semi-supervised, informed, and unsupervised approaches.

A. Semi-Supervised Approaches

The highly variable and distributed nature of IoT networks means that data labeling is an expensive task, requiring human expertise. The aim of semi-supervised approaches is to detect concept drift with the minimal amount of labeled data [10], [11]. Incremental Intrusion Detector (ISF-NIDS) [10] performs unsupervised clustering combined with a supervised cluster adjustment (CA) algorithm, achieving a recall of 85% and false alarm rate (FAR) of 0.9% against the KDD Cup 1999 data set, when using 20% labeled data. Time complexity is given as $O(n)$ for the online clustering phase [10].

Margin Density Drift Detector (MD3) [11], allows for concept drift detection to occur based on unsupervised monitoring of the uncertainty margin of SVM or random subspace of ensemble classifiers, using labeled data only to confirm a suspected concept drift. Combined with an ensemble classification approach, an accuracy of 89%, using 7.9% labeled data was demonstrated with the NSL-KDD data set. Based on the algorithm presented in [11], the time complexity of MD3 is $O(n)$.

B. Informed Approaches

Informed methods make use of a separate change detection algorithm to decide when to collect training data and retrain the model, allowing for faster adaptation to concept drifts [5]. MD3 is an example of a novel informed approach [11]. Informed approaches can be model independent whereby the change detector is separate to the classification model, as demonstrated by the concept drift based ensemble incremental learning intrusion detection (CDIL) method for data streams [4]. Drift Detection Method with Hoeffding's Inequality (HDDM), an SPC based change detector, is combined with master and standby ensemble classifier models, determining when to train and replace the models to adapt to concept drift. This method preserves the models in order to counter a performance degradation problem known as *global replacement*, whereby the model is replaced by one that is trained on only a small number of recent examples [4], [5]. Accuracy is given as 94.9% and recall as 97.22% with the KDD Cup 1999 data set.

An alternative to global replacement strategy is to adopt *local replacement*, whereby only part of the model changes to adapt to new concepts, for example with very fast decision trees (VFDT). Hoeffding Adaptive Tree (HAT) is an example of a VFDT algorithm [12], where the Adaptive Windowing (ADWIN), a detection window based change detection algorithm, is used to determine when to split or remove trees. Accuracy with a custom power station data set was 98% and 94% for binary and multiple classes, respectively.

C. Unsupervised Approaches

Clustering techniques typically require multiple passes over the data set, this is not possible with online learning where data items can be processed only once. To accommodate this, online-offline data stream clustering algorithms are proposed whereby microclustering or grid based methods are applied online before final clustering is applied offline [13].

Density based clustering algorithms are well suite to variable networks, providing the ability to remove noise and form arbitrary shapes to accurately model the ground truth [13]. DenStream uses microclustering online and DBSCAN offline, and was evaluated by [14] using a novel HTTP data set, and [15], using the KDD Cup 1999 data set. True positive rate was reported as 73% [14], and accuracy as 86% [15]. The online time complexity of DenStream is given as $O(q.n)$, and offline as $O(q.logq)$ [13].

D-Stream is a density based algorithm using a grid method online, evaluated by [16], accuracy was reported as 96.5% with the KDD Cup 1999 data set. The time complexity of D-Stream is given as $O(p.n)$ [13].

IV. Online IDS Evaluations

Informed and Unsupervised approaches were evaluated using the Massive Online Analysis (MOA)[1] framework (version 2019.04) [17], using the 10% KDD Cup 1999[2] [18], and UNSW-NB15[3] training [19] data sets. Both data sets were normalised between [0,1] and nominal features converted to binary numeric. Evaluations were ran on a Windows 10 64bit PC with Intel i7 1.8GHz processor and 8GB RAM. All algorithms are set to the MOA default parameters.

The KDD Cup 1999 data set [18] is a popular IDS benchmark, consisting of a synthetic network of UNIX systems, featuring five classes. UNSW-NB15 [19], is a more recent synthetic data set with network attacks generated by the IXIA PerfectStorm tool, featuring 10 classes. Both data sets feature Denial of Service (DoS) as a majority class, although the attacks are focused on IP networks more typical of the IoT transport layer.

A. Informed Evaluation

The following informed algorithms were evaluated:

- *Naïve Bayes (NB)*: A well known, simplistic algorithm based on Bayes Theorem [9] with a time complexity of $O(n)$
- *Drift Detection Method (DDM)*: SPC change detection algorithm, providing warning and change signals [9].
- *Drift Detection Method based on Hoeffding's Inequality (HDDM)*: Similar to DDM but provides performance guarantees with Hoeffding's inequality, time complexity is given as $O(1)$ [20].
- *Hoeffding Adaptive Tree (HAT)*: a VDFT algorithm, using the Adaptive Windowing (ADWIN) detection window change detector to split or remove trees [9].

Results are given in table I. A sliding window size of 1000 was used for all results. NB, where blind, or no, informed change detection was compared against NB using informed DDM and HDDM change detection, and HAT algorithms. Overall the informed NB approach using either DDM or HDDM is recommended offering the optimal balance of accuracy, time and memory performance. Note how blind NB is unable to sufficiently adapt to concept drift or manage memory effectively as demonstrated by its lower accuracy and higher memory usage compared to the informed counterpart, this is illustrated in fig 2. Between interval 200 and 350, performance notably drops, with the HDDM informed approach demonstrating the fastest recovery, it is likely that this performance could be further improved if strategies to compensate for class imbalance with minority classes are adopted [21].

TABLE I
Informed Evaluation Results

Algorithm	Accuracy (%)	Time (s)	Memory (KB)
KDD Cup 1999			
NB (blind)	94.38	27.48	70
NB (DDM)	99.61	24.25	58
NB (HDDM)	99.59	26.22	58
HAT (ADWIN)	99.56	32.34	868
UNSW-NB15			
NB (blind)	31.93	9.30	180
NB (DDM)	77.19	7.84	149
NB (HDDM)	81.88	7.63	147
HAT (ADWIN)	83.03	9.8	1331

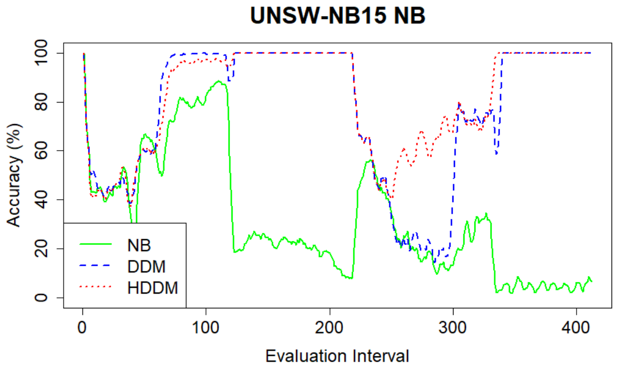

Fig. 2. UNSW-NB15 Naïve Bayes Accuracy

B. Unsupervised Evaluation

The algorithms evaluated were the density based DenStream and D-Stream stream clustering techniques, compared against CluStream which forms micro clusters online and uses k-means clustering offline to form final clusters [13]. CluStream is unable to handle noise and so was used to provide a comparison to the effectiveness of the density based methods [13].

Results are given in table II. The overall performance of both density based techniques is very close, with the clustering purity, the extent to which clusters reflect their majority classes, being lower for D-Stream. As shown in fig. 3, the CluStream algorithm demonstrated the lowest performance, proving that the ability to handle noise and form arbitrary shapes is essential for IDS data streams. It was not possible to collect time and memory metrics in MOA for clustering evaluation due to a known framework problem, confirmed by the project lead[4].

[1] https://moa.cms.waikato.ac.nz/

[2] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[3] https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/

[4] abifet@waikato.ac.nz

TABLE II
UNSUPERVISED EVALUATION RESULTS

Algorithm	**F1 Score (%)**	**Purity (%)**
KDD Cup 1999		
DenStream	92.30	100
D-Stream	92.31	96
CluStream	73.68	97
UNSW-NB15		
DenStream	58.76	82
D-Stream	63.73	76
CluStream	44.22	85

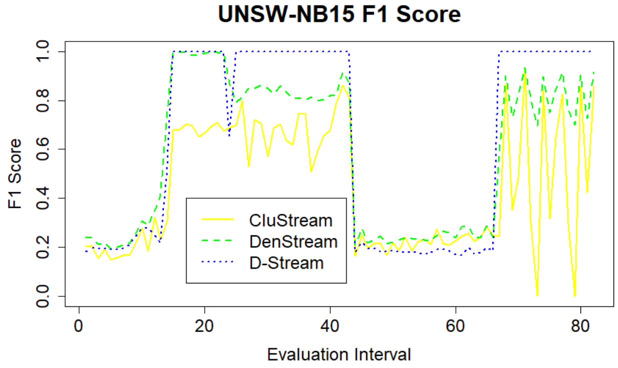

Fig. 3. UNSW-NB15 F1 Score

V. DISCUSSION

Primary challenges of applying intrusion detection to the IoT perception layer are resource constraints, reliability and coverage of a distributed network [1], [2]. The preferred architecture is to deploy distributed IDS sensors within the perception layer to ensure coverage, however this limits the ability to monitor a wide array of attacks and technologies due to invidual sensor resource constraints [1], [2]. Online machine learning techniques can provide a resource efficient architecture for IoT networks in the following ways:

- *Memory Management*: Online learning models process each data item only once, and place priority on recent examples from which new concepts can be learned. Because of this a strategy of abrupt or gradual forgetting can be employed to maintain a constant memory size, which can be tuned with correponding window sizes and fading factors [5].
- *Efficient Learning*: Data labeling is expensive, and impractical for IoT networks [10], [11], use of semi-supervised techniques minimises the requirement for labeled data to detect concept drift. Informed change detection results in models only collecting training data when concept drift is suspected, resulting in lower overall memory requirements as shown in the Naïve Bayes blind vs informed evaluation presented in this paper. Classification can be performed by singular or an ensemble of simplistic algorithms such as Naïve Bayes, where time complexity is linear, requiring less overall processing time.
- *Unsupervised Anomaly Detection*: A key characteristic of an IDS is to perform anomaly detection to detect unknown attacks. The preference would be to combine this in a hybrid approach with misuse detection [3]. This paper has demonstrated the potential of density based algorithms to perform clustering on noisy data streams generated from IoT networks, although further analysis is required into processing and memory metrics. It could be possible to implement a hybrid architecture as suggested in [1] whereby a clustering IDS is deployed on dedicated nodes at key aggregation points within the perception layer in order to perform anomaly detection.

An example architecture is presented in fig. 4. Here a time and memory efficient, informed online learning approach, such as NB with HDDM, is used on IoT nodes to perform misuse detection. Stream clustering techniques are used at key aggregation points to perform anomaly detection, with suspected attacks fed to an alert aggregator, hosted at the application layer. Connection to open wireless networks with limited security features means IoT nodes are inherently unreliable [1]. Discrepancies in results can be identified through the alert aggregation to detect and isolate unreliable nodes.

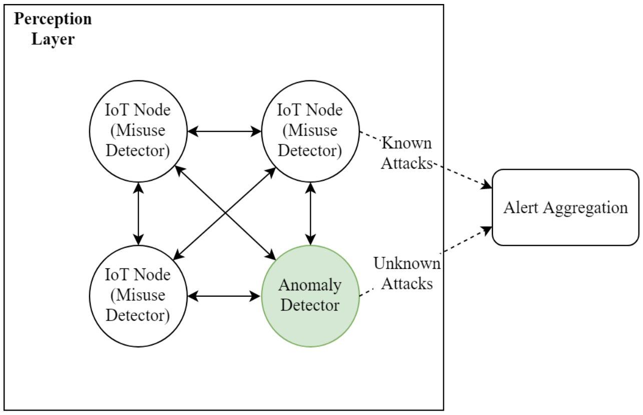
Perception Layer
IoT Node (Misuse Detector)
IoT Node (Misuse Detector)
Known Attacks
Alert Aggregation
Unknown Attacks
IoT Node (Misuse Detector)
Anomaly Detector

Fig. 4. Example Perception Layer IDS Architecture

A key challenge with IDS machine learning is class imbalance. The overall distribution of monitored network data will be heavily biased towards benign traffic, with attack traffic forming minority classes which will be difficult to distinguish from background noise [21]. Strategies to handle class imbalance with incremental online techniques, such as upsampling and ensemble learners, is a proposed area of further research, this is also a key consideration for stream clustering techniques which may wrongly identify attacks as noise, proactively removing them from the model.

VI. CONCLUSION

This paper aimed to answer if online machine learning techniques can provide a resource efficient architecture for IoT intrusion detection. Processing a continuous data stream introduces time and memory constraints that translate to the resource constraints of IoT perception layers. Online learning enables efficient memory management and encourages use of simplistic, time efficient, algorithms for misuse detection. Informed change detection enables models to adapt to changing network conditions overtime, providing adaptation to variations in device and network behaviour and new attacks. Semi-supervised and unsupervised approaches minimise the need for labeled data and provide opportunities to perform anomaly detection for unknown attacks. Class imbalance will degrade performance of minority classes that represent legitimate attacks and is a recommended area for future research.

References

[1] E. Benkhelifa, T. Welsh, and W. Hamouda, "A critical review of practices and challenges in intrusion detection systems for iot: Toward universal and resilient systems," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3496–3509, 2018.

[2] A. Tabassum, A. Erbad, and M. Guizani, "A survey on recent approaches in intrusion detection system in iots," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 2019, pp. 1190–1197.

[3] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[4] X. Yuan, R. Wang, Y. Zhuang, K. Zhu, and J. Hao, "A concept drift based ensemble incremental learning approach for intrusion detection," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2018, pp. 350–357.

[5] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM computing surveys (CSUR)*, vol. 46, no. 4, p. 44, 2014.

[6] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35 365–35 381, 2018.

[7] G. Apruzzese, M. Colajanni, L. Ferretti, A. Guido, and M. Marchetti, "On the effectiveness of machine and deep learning for cyber security," in *2018 10th International Conference on Cyber Conflict (CyCon)*. IEEE, 2018, pp. 371–390.

[8] V. Losing, B. Hammer, and H. Wersing, "Incremental on-line learning: A review and comparison of state of the art algorithms," *Neurocomputing*, vol. 275, pp. 1261–1274, 2018.

[9] A. Bifet, R. Gavaldà, G. Holmes, and B. Pfahringer, *Machine Learning for Data Streams with Practical Examples in MOA*. MIT Press, 2018, https://moa.cms.waikato.ac.nz/book/.

[10] F. Noorbehbahani, A. Fanian, R. Mousavi, and H. Hasannejad, "An incremental intrusion detection system using a new semi-supervised stream classification method," *International Journal of Communication Systems*, vol. 30, no. 4, 2015. [Online]. Available: https://doi.org/10.1002/dac.3002

[11] T. S. Sethi and M. Kantardzic, "On the reliable detection of concept drift from streaming unlabeled data," *Expert Systems with Applications*, vol. 82, pp. 77–99, 2017.

[12] U. Adhikari, T. H. Morris, and S. Pan, "Applying hoeffding adaptive trees for real-time cyber-power event and intrusion classification," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 4049–4060, 2017.

[13] S. Mansalis, E. Ntoutsi, N. Pelekis, and Y. Theodoridis, "An evaluation of data stream clustering algorithms," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 11, no. 4, pp. 167–187, 2018.

[14] D. Stevanovic and N. Vlajic, "Next generation application-layer ddos defences: applying the concepts of outlier detection in data streams with concept drift," in *2014 13th International Conference on Machine Learning and Applications*. IEEE, 2014, pp. 456–462.

[15] S. Ding, J. Zhang, H. Jia, and J. Qian, "An adaptive density data stream clustering algorithm," *Cognitive Computation*, vol. 8, no. 1, pp. 30–38, 2016.

[16] Y. Chen and L. Tu, "Density-based clustering for real-time stream data," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2007, pp. 133–142.

[17] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "MOA: massive online analysis," *J. Mach. Learn. Res.*, vol. 11, pp. 1601–1604, 2010. [Online]. Available: http://portal.acm.org/citation.cfm?id=1859903

[18] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*. IEEE, 2009, pp. 1–6.

[19] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 military communications and information systems conference (MilCIS)*. IEEE, 2015, pp. 1–6.

[20] I. Frías-Blanco, J. del Campo-Ávila, G. Ramos-Jimenez, R. Morales-Bueno, A. Ortiz-Díaz, and Y. Caballero-Mota, "Online and non-parametric drift detection methods based on hoeffding's bounds," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 3, pp. 810–823, 2014.

[21] T. R. Hoens, R. Polikar, and N. V. Chawla, "Learning from streaming data with concept drift and imbalance: an overview," *Progress in Artificial Intelligence*, vol. 1, no. 1, pp. 89–101, 2012.