A New Algorithm Which Runs in Linear Time Enables the Transformation of Legacy Equipment Into Autonomous and Trustworthy IoTs

Ole Kristian Ekseth
Department of Computer Science (IDI)
NTNU
Trondheim, Norway
oekseth@gmail.com

Erik Morset
Winns
Trondheim, Norway
Erik@winns.no

Svein-Olaf Hvasshovd
Department of Computer Science (IDI)
NTNU
Trondheim, Norway
sophus@ntnu.no

***Abstract*—-The time-cost of today's classification algorithms are all too high: the use of existing algorithms makes it impossible for cloud-based systems to provide decision-support for remote sensors. Thus, there is a need to develop new algorithms with sufficient accuracy, and with explainable outcomes. Thereby, enabling improved utilization of industrial/physical equipment through smart control. In this work we address this requirement: this work presents a new methodology for learning, and training, of classification algorithms. The results indicate that the algorithm outperforms existing methods by 10,000x. Importantly, the new algorithm has a memory footprint considerably smaller than similar strategies, and is straightforward to validate for trustworthiness. This makes it possible to deploy the algorithm at both IoTs and in the cloud, thereby ensuring its broad applicability.**

***Index Terms*—Approximate Computing, performance, execution-time, signal and image processing, segmentation and clustering, machine learning, algorithms, correlation, similarity-metrics.**

I. Introduction

Today, there is an increasing focus on autonomous regulation of sensors: in the energy sector, there is a direct link between automated regulation versus the heating bill [1]. However, the shift from systems with a high degree of manual maintenance to automated sensor logic makes the systems vulnerable to penetration attacks [2]. A recent lapse in penetration security led to “the compromise of 1.9 billion records” [3]. Examples of penetration attacks are:

1) malicious firmware upgrades parameters, *e.g.*, to make use of vulnerabilities in the remote device management interface [4], [5];
2) reading of sensitive sensor data [5], [6];
3) manipulation of actuators through compromising raw or processed sensor data [7].

This requires accurate, fast, and trustworthy algorithms. The problem is that existing algorithms for AI can not be used to control many of today's industrial facilities. This is due to the limited processing power of industrial equipment, combined with issues in data bandwidth, and challenges in certifying algorithms for AI. This paper seeks to address this issue through the design of a new model for classification algorithms.

The increased focus on AI has spurred approaches for automated event detection and prevention [8]. The global sensor market is expected to reach $287.00 Billion by 2025 [9]. Suppliers of industrial control systems are subjected to the same technical challenges, as seen for issues in low data throughput [10] and computational cost of algorithms [11], [12]. Hence, addressing issues in data analysis is bound to significantly increase the value of companies addressing this challenge.

This argues for the design of algorithms applicable to legacy *Internet / Intelligence of Things* (IoTs): if we transform existing equipment into autonomous control units (*e.g.*, to control the heating of hospitals), the result is a reduced amount of traffic on low-latency networks, hence reducing the impact of malicious hacking. By proving existing sensor-components with the flexibility of configurations, one reduces the frequency of firmware updates. Through the use of explainable AI, equipment owners (*e.g.*, owners of real estate) get trust in equipment, thus, enabling the certification (and application) of the systems to environments requiring a high degree of uptime. Therefore, if one manages to design a classification algorithm based on these criteria, the result is an increased accuracy of sensors, *i.e.*, without introducing threats to cybersecurity.

To address these requirements, this paper explores a new methodology for construction of AI. The scope of this paper is as follows: Section II outlines the contributions of this paper, Section III evaluates existing strategies for tuning legacy equipment into smart IoTs, Section IV describes a new $O(n)$ algorithm for tuning dumb equipment into smart IoTs (enabling a 10,000x+ reduction in execution-time), Section V evaluates the accuracy and applicability of the guidelines, Section VI relates the findings to requirements of autonomous IoTs, while Section VII summarizes the findings.

II. Contributions by this paper

The paper presents a new algorithm for the learning and training of classification algorithms. This work exemplifies how to apply Approximate Computation without loss in prediction accuracy. The paper describes how the system may be applied to industrial systems. The paper identifies a strategy for Approximate Computing that is generalized for a wider audience. The method seeks to intersect algorithms blind spots with knowledge of usage patterns and the physical properties of the data. From the results, we observe how the result is a framework applicable to devices with low computational power, such as IoT networks for control of energy systems. The paper presents results focused on:

1) Approximate Computing: identify generic strategies to simplify the calculation-steps in algorithms;
2) execution-time: identify an algorithm which correctly classifies data in $O(n)$ time (for a data-set with n points);
3) accuracy and provenance: explore the new $O(n)$ algorithm through the classification of images, signals, and generalized application to the MNIST data.

In the following sections, the above perspectives are outlined. The paper relates the concepts of algorithms shortest-paths, combined with Approximate Computing, and knowledge of algorithms Pareto Boundary, to identify a strategy applicable to legacy IoTs, hence enabling existing systems (*e.g.*, sensors controlling heating-systems) to become autonomous.

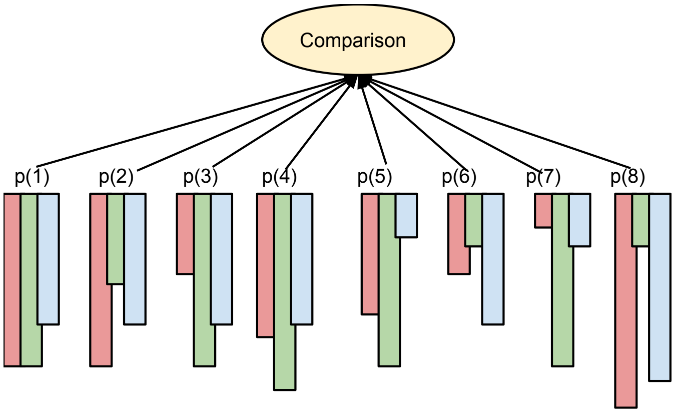

Fig. 1. Why the classification strategy is fast and trustworthy. This figure exemplifies domain-simplifications which makes it possible to reduce the cost of algorithms.

III. Related Work

This paper seeks to identify a strategy for turning legacy equipment into autonomous units. The motivation is to enable autonomous decision-support of infrastructure-critical equipment, for which legacy equipment needs to be updated with new functionality. To reduce the cost of infrastructure (*e.g.*, the heating-bill) there is a need to turn dumb equipment into smart systems, which requires situational awareness of how systems are used [13]. This implies a shift from an analysis of isolated components (*e.g.*, actuators, the pressure of refrigerants, etc.), and into a bird's-eye understanding; if the issues in the performance of classification-algorithms are not resolved, then this task is impossible.

A. Classification Algorithms

There exist numerous clustering-algorithms to use for data-classification:

1) clustering and categorization: a classification algorithm needs to relate clusters of points to an organization reflecting a particular shape (*e.g.*, the shape of letters, the traits of a particular cancer type, etc.);
2) generic algorithms versus domain-application: algorithms are designed towards generic use-cases (*e.g.*, to randomly subdivide points into 'k' number of clusters, as applied in K-means), which results in high error-rate (of generic applications for specialized use-cases);
3) accurate predictions: the specificity of particular use-cases (*e.g.*, interpretation of large versus small feature-differences) makes it important to tailor generic classification-algorithms towards each domain-specific application.

For applications tailored towards IoT and Big Data, the classification algorithm needs to run fast, while having sufficient accuracy. The importance of this requirement is found in how the computers are tightly glued to the physical equipment. A use-case is seen for Winns (a producer of energy systems): Winns reports that the utilization of heating equipment is tightly related to the situational awareness [1].

B. Strategies for Cyber Security in Industrial Systems

When automating infrastructure-critical systems (*e.g.*, heating of hospitals), we need to ensure that the system's behavior (under different circumstances) is correct. Otherwise, the systems can fail spectacularly, as seen at the height of the cold war in 1982 [14]. This involves paying attention to:

1) Penetration Security and System Integrity: certification of algorithm correctness and consistency; application of rules to avoid erroneous changes (in system configurations), handling both intentional and functional configuration-errors;
2) Disaster recovery and business continuity: use of fallback-routines for the handling of system outage;
3) Endpoint protection: Firewalls, Identity and access management (IAM), Intrusion prevention systems (IPS/IDS), Encryption tools, etc.

In this work we focus on addressing these aspects by reducing the expressive power, hence, reducing the number of failed states. This is motivated by the properties of mechanical systems. In mechanical systems there is a limited number of possible combinations, hence, making it necessary to support complex grammar.

C. The design of cyber-secure classification algorithms

The main challenge in the design of accurate and fast heuristics concerns the interception of tacit patterns used by human experts to deduce answers from complex data-sets. This requires a classification algorithm with the following key-features [15], [16]. Therefore, a prerequisite for safe and sound AI algorithm predictions is a strategy for capturing this

ISBN: 978-1-61208-850-1

TABLE I
ASPECTS OF CYBER-SECURITY. THE BELOW TABLE EXEMPLIFIES HOW THE PROPOSED METHODOLOGY ADDRESSES ISSUES IN CYBER-SECURITY AND EXECUTION-TIME.

		Non-Heterogenous AI		Heterogeneous AI	
		Definition		Definition	
What	When	How	Issue	How	Benefit
Data compression	Handling of large data streams	Subsampling, data-reduction, algorithms	Trust	Partial Computing	Known variance
Limited data bandwidth	Communication between devices	Subsampling	Provenance	Partial Computing	Known Variance
Calculation of AI	Data from sensors	Assumptions of distributions	Undocumented assumptions	Semantics	Trust
Configuration of algorithms	Computers with limited resources	Simplified algorithms	Handling of Outliers	Partial Computations	Provenance, Accuracy
Use of reference data in algorithms	State changes	Guesses based on data-changes	Unawareness of changes	Updates through semantics	Pareto Boundary
Handling of Data distortions	Signalling noise in the data-cable	Generalized assumptions	Execution-time, inaccurate assumptions	Semantics, software	Provenance
Classify actors in an image	Object seen from odd angles	Training Data	Loss of inferences	Algorithmic Building Blocks	Cognitive radius
Analysis of external data-sets sampled from different data resources with unknown origin	distortion along an unexpected axis	Average normalization	Unexpected behaviour	Provenance, semantics	Flow of documentation
Analysis never completes	Data too large for microprocessors	Inaccurate algorithm	Inaccurate predictions	hpLysis software	Accurate predictions

intersection: to map cognitive (or: philosophical) perspectives of patterns to the design of fast and accurate algorithms heuristics. This implies addressing the issues of:

1) data throughput: IoT equipment are interconnected through multiple layers of networks with poorer bandwidth, as seen for Fieldbus networks [17];
2) AI-algorithms: a need for accurate and fast classification and ranking of equipment status, which may be generalized into the tasks of cluster analysis for hypothesis evaluation;
3) computing power: computers embedded on IoTs go approx 10^2x slower than the microprocessors found on sensors, and approx 10^4x slower than desktop computers [10].

The observations argue for identifying algorithms that may efficiently be applied to legacy IoTs. If successful, the approach is bound to have an impact on 200 billion+ computers [18]: the global sensor market is expected to reach $287.00 Billion by 2025 [9]. Our earlier research reveals how the cost of AI may be reduced by 100x+ while improving the trustworthiness of predictions [19]. To summarise, the task of redefining existing sensor networks into autonomous equipment requires an AI algorithm with a high degree of prediction trustworthiness and is feasible to integrate on existing Intelligent / Internet of Things (IoT) microprocessors. In the next sections, we outline the results of this strategy.

Algorithm 1 The proposed *ultraFast* algorithm.

```
Output: $clusters = []$
procedure TRAIN(Normalization, MergeMetrics, SimilarityMetrics, EntropyFunctions, $F_e$, $F_s$S)    ▷ Task: learn how to classify data:
    Result    ▷ holds the result-function
    for each $n \in Normalization$ do
        for each $m \in MergeMetrics$ do
            for each $s \in SimilarityMetrics$ do
                vec = []
                for each $f \in EntropyFunctions$ do
                    s = $F_e$(...)    ▷ Task: reduce dimension from $data = [rows, columns]$ to $scalar$
                    vec.push(s)
                ▷ Task: identify accuracy of training-paramters:
                $F_s$(Result, vec, ...)
procedure CATEGORIZE(TrainedData, data)    ▷ Task: Apply the $O(n)$ algorithm:
    class    ▷ Holds the answer
    for each $f \in TrainedData$ do
        t = distance(f, data)
        if ( thent.d < class.d)
            class = t
```

IV. METHOD: A NEW CLASSIFICATION $O(n)$ ALGORITHM FOR ACCURATE CLASSIFICATION

This section describes a framework for construction of a fast classification algorithm (Table I), which involves the design of

TABLE II
THE APPROXIMATE TIME COMPLEXITY OF CLUSTER ALGORITHMS (SUBSET). IN THIS TABLE n DENOTES THE NUMBER OF FEATURE ROWS, f IS THE NUMBER OF FEATURES, c IS THE NUMBER OF CATEGORIES AND I DENOTES THE MAXIMUM ITERATIONS.

[redacted]	Time Complexity:	Relative Time [x] for n=1000	Relative Time [x] for n=1000,000
Proposed: classification (Section IV):	O(n*f)	$1x$	$1x$
KD-TREE [20]:	O(f* n log(n))	$10x$	10^4x
DBSCAN and HP-CLUSTER [21], [22], [23]:	$O(n^2 * f)$	10^4x	10^7x
Hierarchical Cluster Algorithms:	$O(n^2 * f)$	10^4x	10^7x
Kruskals MST [24]:	$O(n^2 * f)$	10^4x	10^7x
K-means [25]:	$O(n^2 * f + I * c * n * f)$	10^4x	10^7x
SOM [26]:	$O(n^2 * f + I * c^2 * n)$	10^4x	10^7x
Neural Networks []:	$O(f * n^5)$	$1000^4x = 10^7x$	$10^{10}x$

an algorithm where:

1) Approximate Computing: subsection IV-A identifies a strategy to transform existing algorithms (which makes use of multiple centroids, or: neurons) into a problem requiring a single neuron;
2) execution-time: subsection IV-B describes an algorithm which turns the observation from subsection IV-A into an $O(n)$ algorithm;
3) accuracy and provenance: subsection IV-C exemplifies how the $O(n)$ algorithm (subsection IV-A) applied to image classification.

A. Problem transformation: application of Least Parsimony to Neural Networks

To reduce the execution time of algorithms, it is of importance to minimize the number of dimensions to evaluate. Hence, to transform the evaluation problem through the principle of Least Parsimony [15]. The idea is to compute entropy by taking the distance from each midpoint to each color. SOM organises the points based on similarities in RGB. The work of [27] applies SOM to construct a two dimensional scheme of entropy computations (Eq. 1):

$$signature = \sum_{x \in C} min(C_k, d(x, C_k))\, x \in C_k \qquad (1)$$

where C_k denotes data-rows in cluster k, $d(x, C_k)$ is the feature similarity between the cluster versus the data-row x, and where C holds the clusters, while $|C|$ holds the set of all data-rows (*e.g.*, in the input image). From Eq. 1 we observe how prediction inaccuracies arise when the *within-distance* is not significantly greater than the *between-distance* (Eq. 2):

$$\sum_{x \in C} min_{k \in C}(min(d(x, C_k))) < \textit{Eq. 1} \qquad (2)$$

For cases where the *between distance* is smaller than the *within distance* the splitting of points between clusters becomes pointless (2), *i.e.*, as the prediction specificity is not improved. Hence, when SOM is applied for data outside the algorithms Pareto Boundary then using multiple centroids (or: neurons) increases the prediction error rate. This exemplifies how costly algorithms may be redesigned into the use of a single reference point, where the latter becomes equivalent to the direct use of entropy metrics.

B. An $O(n)$ algorithm for classification

The motivation is to design an effective algorithm for classification. This algorithmic learning-phase can be generalized into:

1) ensemble data: a list of ranked (*i.e.*, ordered) data; used to determine in the *selection phase* to determine the best-performing *algorithm permutation*;
2) algorithm permutation: uses a selection of building blocks to construct a pipeline of algorithm-training; the iterative sequence (of this feature-scaling) ensures that the identified algorithm has a time complexity of $O(n)$ for n data-points;
3) selection phase: each *algorithmic perturbation* produce a scalar number (*e.g.*, number=2.0001); the number is inserted into a vector; when all the data-sets (in a data-ensemble) are calculated, the vector is compared to the expected order (of data, as defined in the *enamdale data* phase);

The above steps are formalised into an algorithm for value selection (Alg. 1). The following subsection IV-C exemplifies how Alg. 1 can be trained for image-classification, a task of higher complexity than classification of sensor-data.

C. Automated Algorithm Configuration: training and evaluation

To train the algorithm, we provide a tool-suite for the exploration of algorithm combinations, and templates for mapping the properties into implementation with low execution-time and small assembly instruction size. Therefore, the approach may be used for the training of algorithms applicable to legacy IoT microprocessors. An example is to apply an automated evaluation strategy considering the building blocks of:

1) entropy metric: explore 20+ metrics for capturing the variance in a distribution of numbers;
2) down-sampling: condense numbers through compression, for which blocks of adjacent numbers are constructed;
3) blurring: include perspective provided by each number through brushing, for which we explore the combinations of *unchanged*, use a linear attenuation threshold, etc.;
4) strategy for converting input image to histogram: none, bins=[10, 100, 1000] x [raw, average, sum];

5) RGB to scalar conversion: translate the "Red, Green, Blue" scores in images into a singular channel (*e.g.*, Hue);
6) normalization: explore the effect of normalization values through different combinations of the midpoint (*e.g.*, the value of averaged score), signed, etc.;
7) combine data: determine how gold hypothesis is to be used, *e.g.*, to merge features based on relationships such as: multiply, (maximum/minimum), etc.;
8) pairwise similarity metric: apply the 320+ metrics [28].

The results provide a proof of concept for the assertion that entropy-algorithm supports *re-invention*, *e.g.*, that it manages to get results at-least-as-good as the SOM-method. The comparison of data with a known topology avoids the need for complex iteration steps (Figure 1), which is in contrast to other algorithms(eg as "SLINK" [29], "k-means" [25], etc.). Hence, explaining why the proposed framework enables a reduction in execution time by 10^4+ for data-set with 1000 points (Table II).

V. Results

The findings provide insight into the feasibility of transforming Neural Networks into the Ultra-fast $O(n)$ algorithms (Alg. 1). The idea (which is explored) is to use a singular centroid to capture complexities (subsection IV-A), for which a problem is rewritten through use of algorithmic building blocks. The proposed algorithm enables fast and secure communication over insecure networks: by reducing the processing-time, and amount of data to transfer, users are able to apply cryptography strategies (an overhead would otherwise be unbearable). To validate the feasibility of the proposed guidelines we explore:

1) accuracy and provenance: to measure the algorithm's feasibility, we investigate the algorithm through generic, and specific, use-cases, *e.g.*, the accuracy of image-classification;
2) execution-time: to identify any overhead in execution-time, the hpLysis software [30] is updated with Alg. 1, where results are summarized in Table II;
3) approximate computing: to explore the effects, this paper transform a set of complex classifications into a simple comparison (Figure 1) through the use of Alg. 1 (Eq. 1), and then explores the difference in performance.

The results are summarized in Table II, which identifies the relative execution-time for different algorithms. Winns (a producer of heat-exchange pumps [1]) reports that sensor-predictions need to be returned in less than 10 seconds, which only Alg. 1 manages (Table II). When Alg. 1 is evaluated through the above perspectives we observe how Alg. 1 outperforms the base-line algorithms in use:

1) Specific application: classification of image-data, here exemplified through the Las Vegas data-set and the Lake Mead dataset found in [31];
2) Signal classification: classify shapes with different growth-ratios and ranomdation, *i.e.*, $y(r, a, x) = r_1 a_1 x^0 + r_2 a_2 x^2 + ... r_n a_n x^n$, where $y(..)$ is the feature-vector to evaluate, n is the number of combinations (to construct a signal from), r is a constant randomization-factor, a is a constant attenuation-factor (*e.g.*, $a = 1.5$), and x represents the polynomial variabel-part;
3) Generalized applicability: the hpLysis is updated with generalized tests, each investigating the effects of Approximate Computing on Neural Network.

To exemplify, a comparison between SOM-strategy for low-latency classification (undertaken by [27], [31]) versus Alg. 1 (as proposed in Section IV) indicates the transformation of algorithms into using a singular centroid (Eq. 1) can substantially boost the performance of analytical approaches. Discussions with the authors of [31] reveal how data-specific configurations of the SOM are required to get the algorithm to produce correct results. The results reveal how the use of a singular centroid (in data-classification) provides a simple, yet effective, strategy for trustworthy control of classification-tasks. Therefore, the algorithm may readily be used on existing sensor networks, *e.g.*, to control equipment for heat-exchange in buildings.

VI. Discussion

This paper has identified a low-cost method to classify data with well-defined characteristics, which is the case for sensors that monitor physical equipment (Table I). To reduce the scope, the paper focuses on industrial control-systems which a) are sensitive to delays in configurations, and b) where certification of behavior represents a crux. The paper argues that a holistic perspective of classification-algorithms results in a cost-effective strategy to address issues in data-throughput. The proposed methodology, and algorithm, differ from the established strategy. To exemplify:

1) Approximate Computing: this work explores the benefit of closely gluing compiler-optimization with the accuracy of algorithms, *e.g.*, in contrast to "scikit learn" [32];
2) execution-time: we transform complex algorithms (into their simplified counterparts) by merging the cluster-centroids, *e.g.*, in contrast to [31];
3) accuracy and provenance: the use of metric-training (Alg. 1) a) relates to a system's physical properties, and b) captures the algorithmic behavior, *e.g.*, in contrast to [33].

This work exemplifies a methodology that is generalizable for a wider audience; through an intersection between established algorithms, use-cases, and configurations, the paper reveals a strategy reducing the execution-time by more than 10,000x. The paper argues that the approach can be applied to arbitrary cases of classification, such as the classification of sensor data from IoT, a ranking of satellite images, etc. A concrete example concerns the effects of the accurate choice of pairwise similarity metrics in the clustering algorithm.

VII. Conclusion

The paper proposes a parametric strategy to increase the applicability of classification algorithms: observations relating to the approximate nature (of classification algorithms) are used to derive a new $O(n)$ algorithm. The algorithm is both evaluated through generalized, and highly specific, datasets, hence ensuring its broad applicability. The use of well-defined metrics, reflecting the physical properties of sensor-systems, makes the algorithm easy to certify: the seamless use of off-the-shelf building blocks address issues in data-throughput, the trustworthiness of predictions, and the speed of microprocessors, *i.e.*, without resulting in increased component costs. Thereby, the paper provides a template addressing the daunting challenges facing researchers, managers, and owners of industrial systems. Hence, the proposed algorithm addresses the conceptual challenges which currently hampers the development of trustworthy applications of AI to the autonomous control of industrial systems.

The findings presented in this paper indicate the need for updating the requirements for the certification of sensors and equipment. Hence, there is a need for a concerted effort in the industry, *i.e.*, to devise a formal protocol that ensures flexible and safe AI for industrial sensor networks.

REFERENCES

[1] E. Morset, “Email conversations with the cto of winns reveals how accurete regulations of heat-pumps maps to their energy consumption.” Winns, 2021, accessed: January 2021.

[2] S. Liu, B. Xing, B. Li, and M. Gu, “Ship information system: overview and research trends,” International Journal of Naval Architecture and Ocean Engineering, vol. 6, no. 3, 2014, pp. 670–684.

[3] E. McMahon, M. Patton, S. Samtani, and H. Chen, “Benchmarking vulnerability assessment tools for enhanced cyber-physical system (cps) resiliency,” in 2018 IEEE International Conference on Intelligence and Security Informatics (ISI). IEEE, 2018, pp. 100–105.

[4] US-CERT, “Alert (ta16-288a) heightened ddos threat posed by mirai and other botnets,” 2016, accessed: September 2019. [Online]. Available: https://www.us-cert.gov/ncas/alerts/TA16-288A

[5] K. Q. Ye, M. Green, N. Sanguansin, L. Beringer, A. Petcher, and A. W. Appel, “Verified correctness and security of mbedtls hmac-drbg,” in Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2017, pp. 2007–2020.

[6] J. H. Ziegeldorf, O. G. Morchon, and K. Wehrle, “Privacy in the internet of things: threats and challenges,” Security and Communication Networks, vol. 7, no. 12, 2014, pp. 2728–2742.

[7] B. Xing, J. Dai, and S. Liu, “Enforcement of opacity security properties for ship information system,” International Journal of Naval Architecture and Ocean Engineering, vol. 8, no. 5, 2016, pp. 423–433.

[8] D. Trendafilov, K. Zia, A. Ferscha, A. Abbas, B. Azadi, J. Selymes, and M. Haslgrübler, “Cognitive products: System architecture and operational principles,” 2019.

[9] Bloomberg, “Sensor market estimated to reach 287 billion globally by 2025,” 2019, accessed: September 2019.

[10] T. Adegbija, A. Rogacs, C. Patel, and A. Gordon-Ross, “Microprocessor optimizations for the internet of things: A survey,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 37, no. 1, 2017, pp. 7–20.

[11] O. K. Ekseth, M. Gribbestad, and S.-O. Hvasshovd, “Inventing wheels: why improvements to established cluster algorithms fails to catch the wheel,” in The International Conference on Digital Image and Signal Processing (DISP19), Springer, 2019.

[12] O. K. Ekseth, J. C. Meyer, and S. O. Hvasshovd, “hplysis database-engine: A new data-scheme for fast semantic queries in biomedical databases,” in Semantic Computing (ICSC), 2018 IEEE 12th International Conference on. IEEE, 2018, pp. 383–390.

[13] L. Ana and A. K. Jain, “Robust data clustering,” in Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, vol. 2. IEEE, 2003, pp. II–128.

[14] T. Economist, “War in the fifth domain: Are the mouse and keyboard the new weapons of conflict?” 2010, accessed: June 2020. [Online]. Available: https://www.economist.com/briefing/2010/07/01/war-in-the-fifth-domain

[15] B. Dresp-Langley, O. K. Ekseth, J. Fesl, S. Gohshi, M. Kurz, and H.-W. Sehring, “Occams razor for big data? on detecting quality in large unstructured datasets,” Applied Sciences, vol. 9, no. 15, 2019, p. 3065.

[16] O. K. Ekseth, P.-J. Furnes, and S.-O. Hvasshovd, “Pattern matching in the era of big data: A benchmark of cluster quality metrics.” International Journal On Advances in Software, 2019.

[17] A. Pietak and M. Mikulski, “On the adaptation of can bus network for use in the ship electronic systems,” Polish Maritime Research, vol. 16, no. 4, 2009, pp. 62–69.

[18] Intel, “200 billion iot devices in 2020, a market values to $6.2 trillion in 2025,” 2019, accessed: December 2019. [Online]. Available: https://www.intel.com/content/dam/www/public/us/en/images/iot/guide-to-iot-infographic.png

[19] O. K. Ekseth and S.-O. Hvasshovd, “An empirical study of strategies boosts performance of mutual information similarity,” in International Conference on Artificial Intelligence and Soft Computing. Springer, 2018, pp. 321–332.

[20] J. H. Friedman, J. L. Bentley, and R. A. Finkel, “An algorithm for finding best matches in logarithmic expected time,” ACM Transactions on Mathematical Software (TOMS), vol. 3, no. 3, 1977, pp. 209–226.

[21] M. Ester, H.-P. Kriegel, J. Sander, X. Xu et al., “A density-based algorithm for discovering clusters in large spatial databases with noise.” in Kdd, vol. 96, no. 34, 1996, pp. 226–231.

[22] O. K. Ekseth and S. Hvasshovd, “hplysis dbscan: How a memory-aware db-scan implementation out-perform simplified/heuristic db-scan approaches by 1,000,000x+,” 2017, pp. 1–6.

[23] O. K. Ekseth and S.-O. Hvasshovd, “hp-cluster: A new algorithm enables increased performance for clustering of big, and complex, data,” 2020, manuscript ready for submission.

[24] J. B. Kruskal, “On the shortest spanning subtree of a graph and the traveling salesman problem,” Proceedings of the American Mathematical society, vol. 7, no. 1, 1956, pp. 48–50.

[25] S. Lloyd, “Least squares quantization in pcm,” IEEE transactions on information theory, vol. 28, no. 2, 1982, pp. 129–137.

[26] T. Kohonen and P. Somervuo, “Self-organizing maps of symbol strings,” Neurocomputing, vol. 21, no. 1, 1998, pp. 19–30.

[27] J. M. Wandeto and B. Dresp, “Ultrafast automatic classification of sem image sets showing cd4 + cells with varying extent of hiv virion infection.” International Journal On Advances in Software, 2019.

[28] O. K. Ekseth and S.-O. Hvasshovd, “How an optimized DBSCAN implementation reduce execution-time and memory-requirements for large data-sets.” International Journal On Advances in Software, 2017, pp. 321–332.

[29] R. Sibson, “Slink: an optimally efficient algorithm for the single-link cluster method,” The computer journal, vol. 16, no. 1, 1973, pp. 30–34.

[30] Ekseth, Ole Kristian, “hpLysis: a high-performance software-library for big-data machine-learning,” https://bitbucket.org/oekseth/hplysis-cluster-analysis-software/, online; accessed 06. Jan. 2021.

[31] J. M. Wandeto and B. Dresp-Langley, “The quantization error in a self-organizing map as a contrast and colour specific indicator of single-pixel change in large random patterns,” Neural Networks, vol. 119, 2019, pp. 273–285.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg et al., “Scikit-learn: Machine learning in python,” Journal of Machine Learning Research, vol. 12, no. Oct, 2011, pp. 2825–2830.

[33] D. Moulavi, P. A. Jaskowiak, R. J. Campello, A. Zimek, and J. Sander, “Density-based clustering validation,” in Proceedings of the 2014 SIAM International Conference on Data Mining. SIAM, 2014, pp. 839–847.