Advancing Sensor Fusion with Semantic Folding: A Case Study on Municipal Water Pump Monitoring

Author: Francisco Webber, Cortical.io AG, Vienna

Abstract

This paper explores the application of Semantic Folding [1], a representation learning method, to sensor fusion challenges, specifically focusing on a municipal water pump monitored by over fifty sensors [2]. Sensor fusion, integrating data from multiple sensors to yield a comprehensive understanding of system states, faces significant complexities and computational demands, especially when implemented in real-time embedded systems. Semantic Folding addresses these issues by transforming sensor data into semantic fingerprints—compact, structured representations that reflect the system's status at any given moment. Through a detailed case study, we demonstrate how these semantic fingerprints simplify the integration process, facilitate real-time processing, and enhance system reliability and predictive maintenance capabilities.

Semantic Folding's utility extends beyond specific system complexities or sensor varieties, suggesting a broad applicability across various domains. The method transforms sensor data into semantic fingerprints that represent the system's current state, facilitating preemptive anomaly detection and system characterization without extensive user input on hyperparameters. Future research is recommended to optimize the Semantic Folding protocol and expand its analytical capabilities, potentially incorporating these fingerprints as feature vectors in machine learning for predictive analysis of complex entities like biological organisms, social structures, and electronic circuits.

The proposed framework aims to redefine traditional sensor fusion approaches by integrating scalable, flexible semantic analysis, potentially transforming real-time monitoring and predictive maintenance strategies across industries.

1. Introduction

Sensor fusion is pivotal in fields like Automotive Systems, Robotics, Healthcare, Mobile Devices, Aerospace, Defense, Security, Industrial Automation or Environmental Monitoring. [3], [4]

Feature-based sensor fusion is a technique where features extracted from different sensor channels are combined to create a unified, richer, more informative representation that can be used for further processing and decision-making. This approach is prevalent in many applications across various fields. The main motivations for employing feature-based sensor fusion include:

1. **Improved Accuracy**: By integrating features from multiple sensors, the fused data can provide a more accurate and robust estimate of the state of the environment, or the system being monitored. This is particularly important in situations where individual sensors might have inherent limitations or be susceptible to noise and errors. [5], [6], [7]
2. **Redundancy and Reliability**: Multiple sensors can offer redundant information, which enhances the reliability of the system. If one sensor fails or provides erroneous data, the system can rely on data from other sensors to maintain performance and accuracy. This redundancy is crucial in critical applications like automotive safety and aerospace. [8], [9], [10], [11]
3. **Complementary Information**: Different sensors capture different aspects of the environment. Combining these complementary features can provide a more comprehensive understanding of the system. With a comprehensive set of sensors, a representation of the system as a whole – a system status – can be generated. [12], [13], [14], [15]
4. **Enhanced Capability for Complex Systems in Realistic Environments**: Complex environments, such as urban settings or natural terrains with varying weather conditions, can

challenge the capabilities of a single type of sensor. Feature-based sensor fusion allows for more robust performance in diverse and challenging conditions. [16], [17], [18], [19], [20]

5. **Improved Processing Speed and Efficiency**: Processing features instead of raw data from sensors can be computationally more efficient. This efficiency is crucial for real-time applications where decisions need to be made quickly, such as in autonomous driving. [21], [22], [23], [24], [25]
6. **Scalability and Flexibility**: Feature-based fusion allows for scalability in system design. New sensors and features can be added as they become available or as needs evolve without redesigning the entire system. [26], [27], [28], [29]

2. Background and Related Work

The foundational concepts of Semantic Folding originate from text data analysis, where it is used to capture the semantic properties of textual data within representations that are characterized by geometry and where similarity is computed using set-theory.

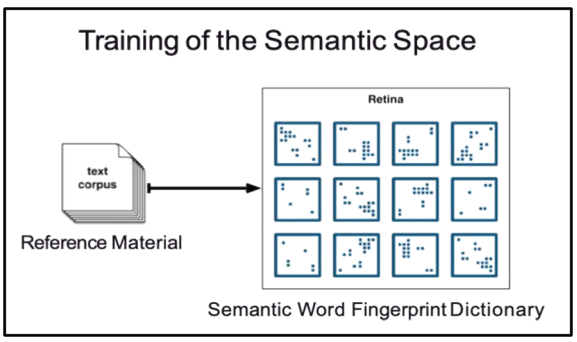
In Semantic Folding three fundamental items are considered:

- Words are the fundamental element of semantics (smallest unit carrying meaning).
- Sentences are the smallest unit representing a concept.
- Texts are sets of concepts and constitute a context.

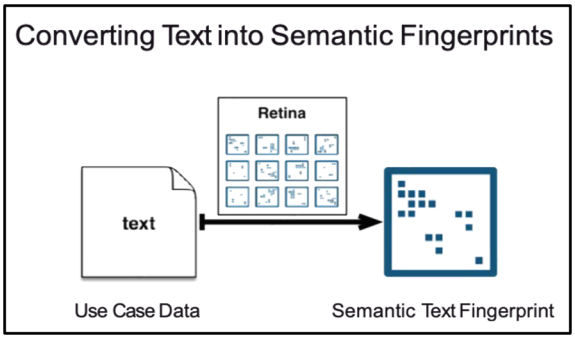
Words are defined by the set of contexts they appear in. During the unsupervised training step text snippets (contexts) are compiled and form the semantic grounding of all word definitions covered by the overall vocabulary.

This collection of reference texts is distributed over a 2-dimensional metric space, associating a specific position with each reference text depending on the word similarity in comparison with all other contexts presented. In a next step the list of all words occurring in the whole reference set is generated. This constitutes the semantic space (Retina) vocabulary.

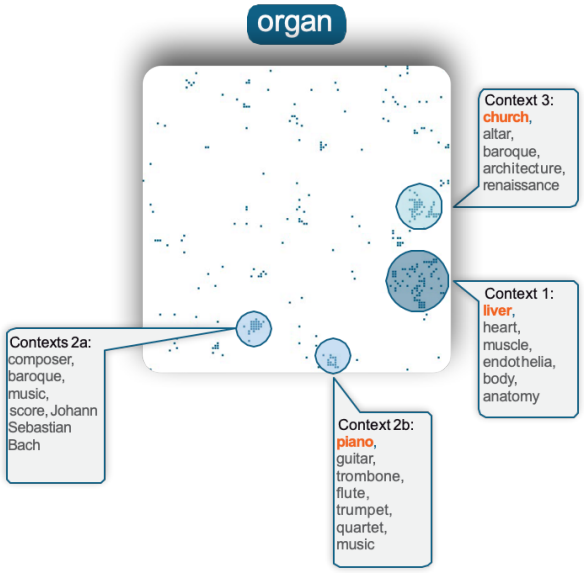
For each vocabulary entry all positional coordinates where the word occurs in a context are retrieved. The set of retrieved positions constitutes the distributed representation for that word [30], [31], [32], [33], [34], in this specific semantic space. After sparsification (2% filling) it becomes a Binary Sparse Distributed Representational Vector [35], [36], [37], [38], [39], [40], [41], [42].



Based on the trained semantic space, any given text can be decomposed into its constituent words, each of which can be converted into its semantic fingerprint representation.

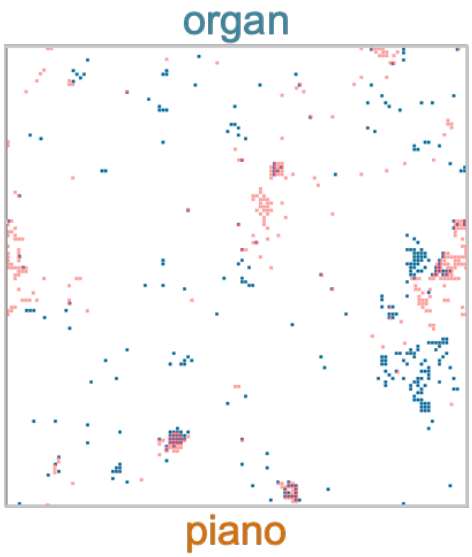


In the following illustration the semantic fingerprint of the English word “organ” is rendered. The size of the semantic topology is set to 128x128, and the reference set used to train the semantic space is a selection of 400K Wikipedia pages.

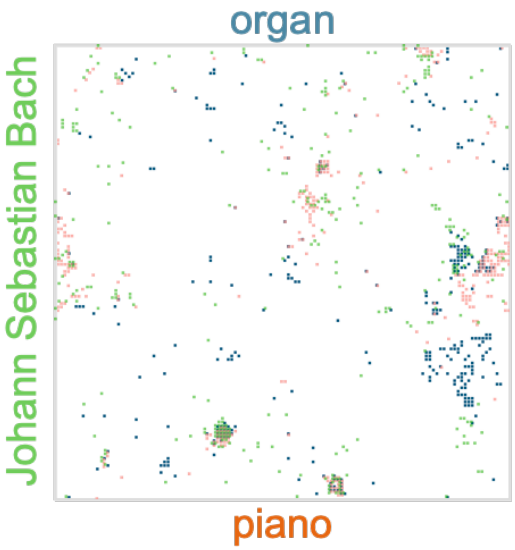


Sparsity is set to 2%, which means a maximum of 320 positions out of 16384 can be active at any given time. Behind every activated position the occurring words can be accessed as context terms. It can be observed that the word “organ” is composed of several context clusters:

- Context 1: This is one sense for “organ” in the context of liver, heart, muscle ..., anatomy → The biological sense of the word “organ”.
- Context(s) 2: Another sense for “organ” → organ the music instruments.
    - Sub-Context 2a: The musical instrument “organ” in context of baroque music.
    - Sub-Context 2b: The musical instrument “organ” in context of other musical instruments.
- Context 3: The role of musical “organs” in relation to churches.
- Context(s) n...: Other less dominant, inspectable, contexts for the word “organ”.



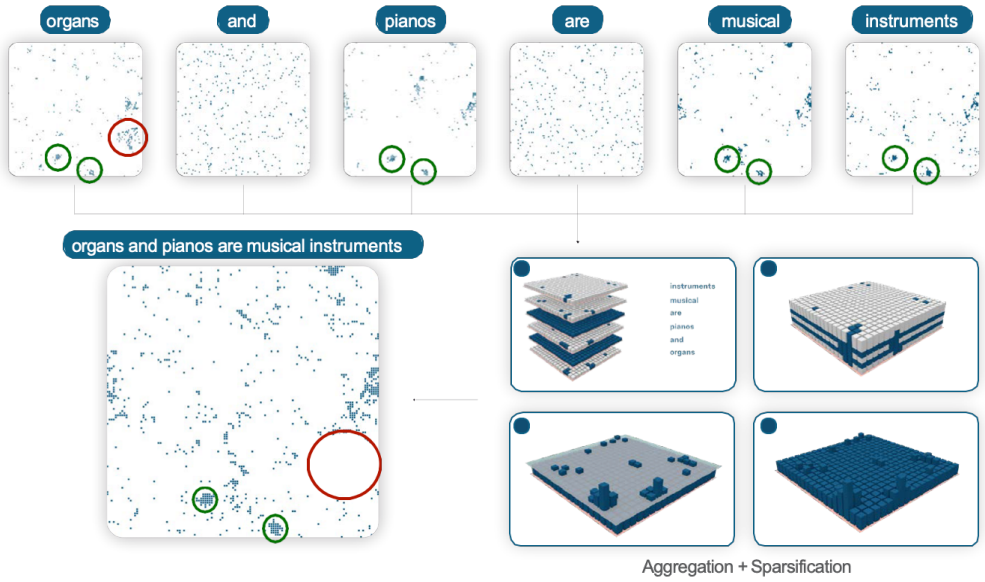
The semantic similarity between the word “organ” and the word “piano” is calculated by overlaying the two words and counting the number of common bits. They strongly correlate in the contexts 2a, 2b and 3 leading to an overall overlap of around 30%.



If not the entire fingerprint, but only the semantic fingerprint of “Johan Sebastian Bach” is used as sub-space (mask) for comparison, it can be noted that the resulting overlap of “organ” and “piano” is scaled to 85%.

A major shortcoming of traditional representational methods is that they must be trained separately for words, sentences, or paragraphs. It is therefore not possible to directly measure the similarity between a word and a sentence as they do not share the same model (set of features).

In Semantic Folding the basic (precalculated) representations are generated for words, more precisely even for tokens like “New York” or “Secretary General of the United Nations”. As these token fingerprints are sparse vectors they can unconcernedly be aggregated (Boolean union) [43], [44], [45], [46], [47] into higher order semantic fingerprints of sentences, paragraphs, documents, books, and even whole libraries.



As a convenient side effect, the resulting aggregated semantic fingerprint has all ambiguous word senses removed as only correct contexts will be maintained after re-sparsification.

Semantic Folding has been applied in a wide variety of use cases and research works. See the list of references [48] to [77] as examples.

By conceptual analogy we extend this method to numerical data, presenting a novel approach to sensor data integration that overcomes the limitations of conventional sensor fusion methods.

3. Semantic Folding and Sensor Fusion

Extensive literature supports the notion that the neocortex employs a similar data processing approach for all inputs, irrespective of their sensorial origin [78-86]. This universal processing mechanism enables the brain to interpret sensory information effectively, helping to construct a coherent understanding of the world. Such capability is the result of evolutionary adaptations aimed at maximizing the sensory system's ability to decode environmental semantics. Semantic Folding, which models these neocortical constraints, can be generalized as follows:

- Semantic Folding is a way to capture the inner state of a system based on a set of perceivable external features. The inner state corresponds to a semantic interpretation of the system, allowing to understand its current state and to predict what behavior/state change can be expected in the future.
- A system consists of a set of elements that can coherently discriminated. These system elements have a state, characterized by some associated feature metric. All elements in a system are interdependent which means that a state change of one element affects all others within the same system.
- The perceivable (radiating) features (in contrast to the inner – hidden – features) of the system are concurrently sampled by the sensors. The set of all sampled features at a given point in time constitute the context for each of the individual features.

This generalized specification can be applied to various types of inputs (beyond its initial special case in natural language) by making an appropriate analogy.

3.1 The Automotive Analogy

Modern cars contain a multitude of sensors, monitoring all important subsystems like the engine, the electric system, the lightning, the driver's cabin etc. All these measurements are relayed to an onboard computer, which subsequently determines any potential next steps. Typically, one can expect to

receive at least 50-100 measurements per second. With autonomous vehicles, it's likely that the number of sensor streams will increase even further.

The on-board computer “interprets”, for instance, the engine temperature as a specific value such as “140” meaning a temperature of 140 degrees Celsius measured by sensor #5 in the engine block. The only thing the board computer can reliably do is to compare the measured values with the maximum permissible temperature specified. Only when some threshold is actually reached, a specifically crafted failure procedure is triggered.

3.2 System State

The only information the monitoring process receives from its sensor is the scalar 140, which limits its ability to react adaptively. As a result, the only recourse is often a drastic reaction once a value exceeds the permissible maximum (or minimum). Consequently, it is difficult to implement context dependent reaction protocols, as these would have to be implemented as discrete state machines, implying a lot of planning, implementation, and testing effort. If the regular operating temperature is expected to be 120° Celsius, then a measurement of 140° could have one or several causes:

- The car is currently driving at high velocity (higher than usual).
- The car is driving up-hill.
- The ambient temperature is elevated.
- Strong headwind could be present.
- A high payload (cargo, caravan) is present.
- etc.

In none of these cases the value of 140° would be “unexpected”. However, if the ambient temperature is low, there is no headwind and the vehicle is driving downhill without passengers, yet the engine temperature is rising to 140°, then the car status is confronted with an anomaly. Like in language, the context, namely all other concurrently sampled values, defines the meaning of a measured value.

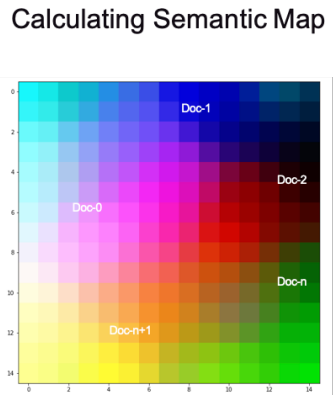
3.3 Capturing the System State

To actually implement the Semantic Folding analogy, the following steps are necessary:

- To generate a Retina (semantic space), it is imperative to establish a reference collection. In linguistic applications, this entails compiling a comprehensive corpus of valid, real-world utterances tailored to the specific use case. Similarly, for automobile sensor fusion, a data stream from the car's sensors is recorded under all anticipated driving conditions—day, night, varying weather conditions (sunshine, rain, snow), and diverse environments (rural areas, inner city, highways, parking garages). Throughout these driving sessions, sets of concurrent sensor values are captured every second and stored in a time-series file. This file serves as the foundational training material for the Retina.
- In this driving log, each line functions as a "document" encapsulating all metrics at a given instant. Each column within these documents represents a specific sensor, and the total number of columns reflects the number of sensors sampled concurrently. Conceptually, each line document can be likened to a "context," analogous to a single sentence in textual data.

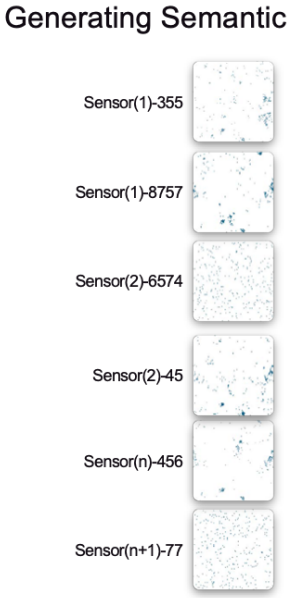
	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor n	Sensor n+1	
t=0	34345	6574	4567	456	874	6578	Doc-0
t=1	355	7664	345	9854	1543	56	Doc-1
t=2	8757	45	166	768	456	9676	Doc-2
t=n	12	443	66556	766	4648	9	Doc-n
t=n+1	198	5778	78	5998	671	77	Doc-n+1

- In this schema, each specific measured value, such as “140” representing 140°C of the motor block, is treated as a distinct "word." Similarly, a measurement of "141" from the same sensor would also constitute a different "word." This analogy equates sensor readings to words in a text, emphasizing the distinct identity of each measurement even when it originates from the same sensor.
- Upon completing the Retina training process with the logged data, each sensor document is assigned a specific position within the fingerprint map. In the metric space of the fingerprint, the arrangement of sensor documents is structured such that documents with similar sensor



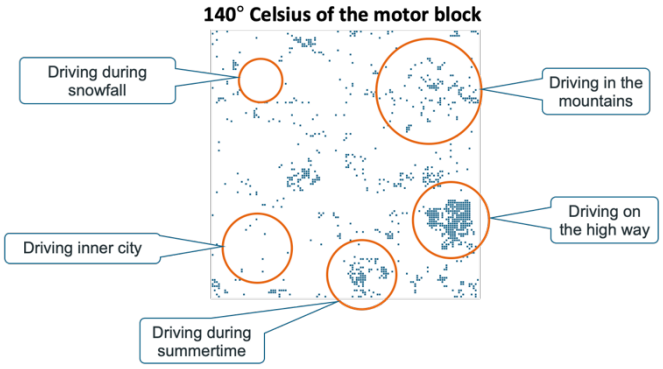
records (Euclidian distance of the document vectors) cluster closely together, while those with greater disparities in measurements are positioned at proportionally greater distances. This spatial organization within the metric space effectively highlights the similarity and dissimilarity among measurement records. The clustering process will group all records originating from similar driving conditions—such as inner-city driving, highway driving, mountain driving, or parking—into distinct clusters.

- Once the distribution of sensor documents is established, the "list of all words" is compiled. In the automotive context, a "word" represents a discrete scalar value from a specific sensor

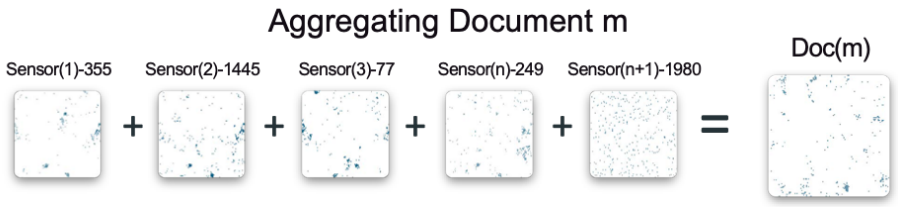


(channel). The actual fingerprint is created by marking the positions on the map with a "1" where the corresponding value appears in any document. This method allows for the generation and storage of all potential semantic fingerprints within the fingerprint (Retina) dictionary.

- The annotated semantic fingerprint for the value 140° Celsius of the motor block could now look like this:



- The fingerprint of sensor #5 value of 140, now serves as a semantic representation that simplifies the assessment of the vehicle's status. This more elevated temperature commonly



manifests when the car is operating under conditions such as mountain driving, highway travel, or in summer weather, indicating specific environmental impacts on vehicle condition.
- The operational condition of the car can be ascertained by consolidating all individual measurement fingerprints from the current sensor document into a unified compound fingerprint. For example, encountering a motor block temperature of 140° Celsius during a downhill drive in winter would result in a compound fingerprint that is clearly identified as an anomaly by displaying unexpected cluster combinations.

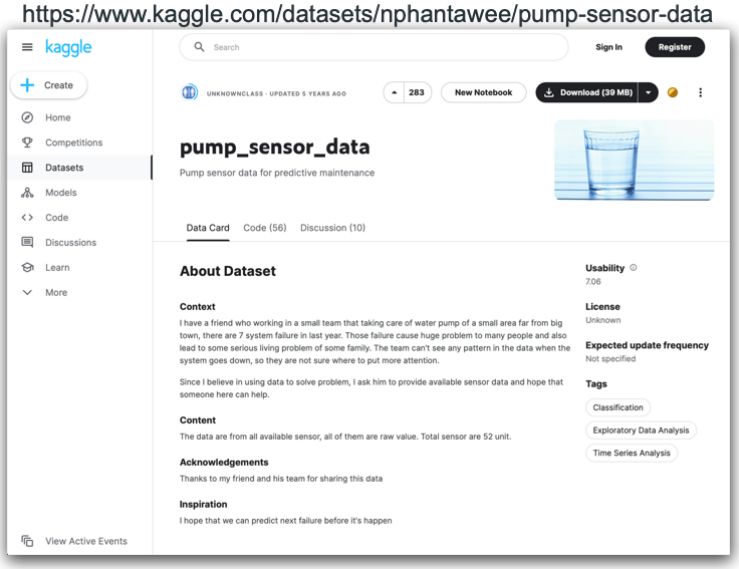
3.4 Predicting Future States

During the "training-drive," capturing actual failures or synthetically generating sensor documents for specific defects enables the creation of a template fingerprint for each defect. When a status fingerprint signals an anomaly, as previously described, it can be compared to a catalog of diagnostic defect template fingerprints. This comparison helps determine the specific cause of the anomaly, enhancing the accuracy and efficiency of diagnostics as well as making a more informed decision on how to proceed with the anomaly.

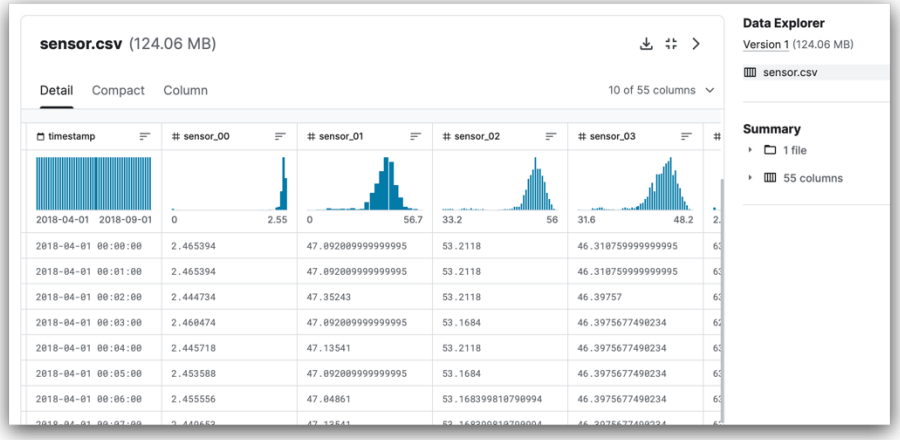
Given the minimal computational demands, anomaly detection and resolution can occur in real time, even on edge processing platforms with limited computational capabilities. Moreover, since the overlap between the status and the diagnostic fingerprints may evolve over time, this incremental anomaly detection can be leveraged for predictive maintenance, allowing for early intervention before failures become critical. This approach enhances both the reliability and efficiency of maintenance strategies.

4. Case Study: Municipal Water Pump Monitoring

https://www.kaggle.com/datasets/nphantawee/pump-sensor-data

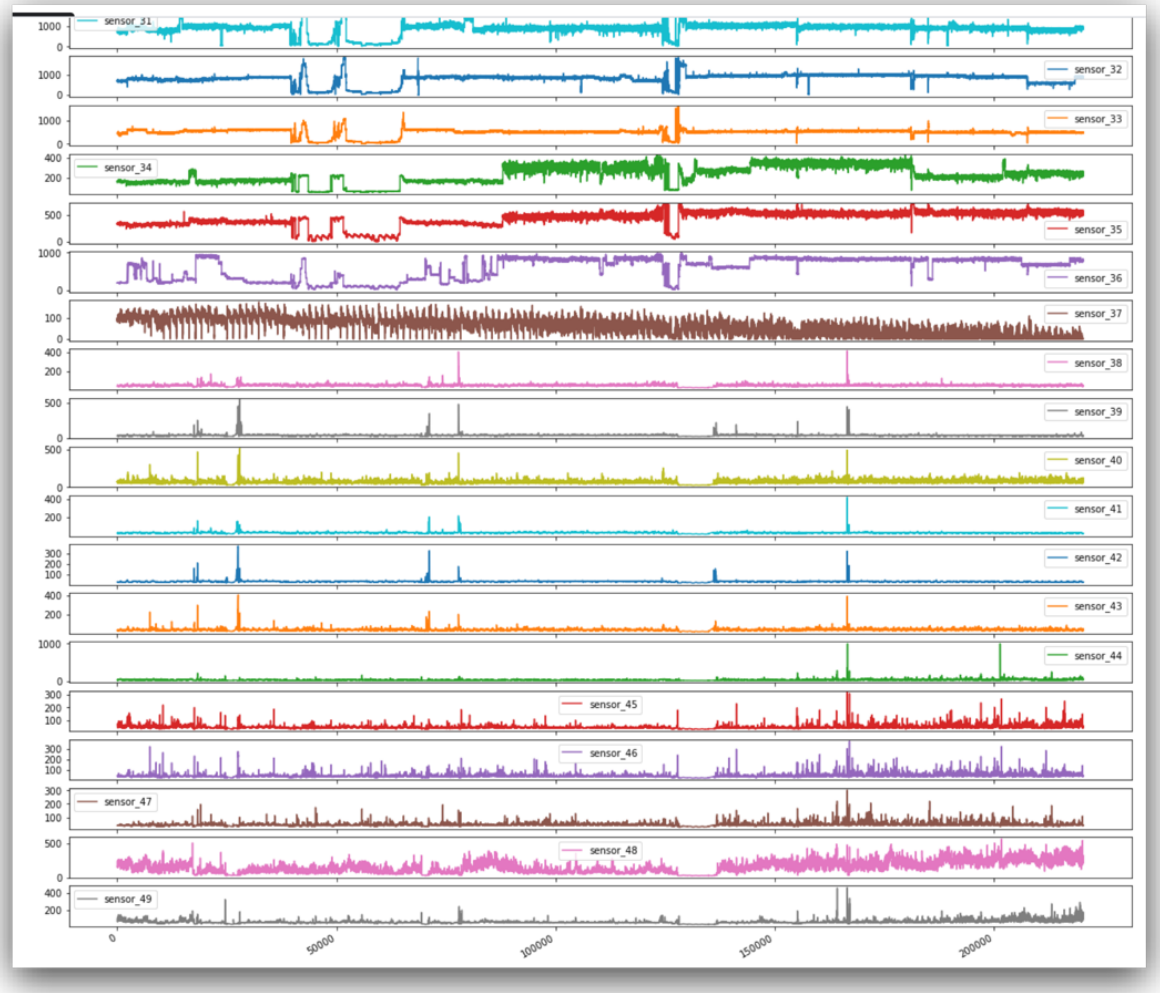


Addressing the sensor fusion problem at its core, necessitates a realistic dataset, which precludes the synthetic creation of data due to the complexity of the hidden semantic information needed.



For this reason, a real-world challenge hosted on the public Kaggle platform was selected to serve as a proof of concept. The challenge centers around a municipal water pump equipped with over fifty sensors. It involves analyzing a time series of measurement points, collected minute-by-minute over four months, which includes data from periods during which the pump experienced failures—specifically seven "BROKEN/RECOVERING" periods. This dataset serves as the foundation for developing a predictive model that aims to anticipate future failures solely based on sensor readings.

The effectiveness of the predictive model hinges on selecting a data representation format for the training vectors that captures and accurately represents the maximum amount of semantic information about the internal state of the pump. The prediction accuracy of the future model is closely linked to the number of available training examples: poorer representations will require more data. Given that the available dataset comprises 220 thousand data points with only a couple of hundred anomaly records —a relatively small number for training deep learning type models—it is crucial that the data representation is both precise and comprehensive to ensure effective learning and prediction.



The initial step in applying the Semantic Folding method involves constructing a reference dataset from data that reflects the water pump's correct, intentional functioning. This dataset includes all records labeled as "NORMAL," while excluding those marked as "BROKEN" and "RECOVERING." The next phase is pre-processing, which involves normalizing the data by determining the smallest and largest values for each sensor and dividing these into predefined bins. Specifically, we utilize 8-bit wide bins, leading to a categorization into 255 distinct ranges. The lowest and highest measurable values are assigned to range-0 and range-255, respectively. Each sensor channel is prefixed with a letter code ranging from A to AZ before the range number to categorize the data effectively.

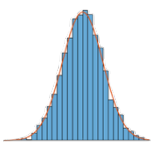
- Input as CSV File

```
,timestamp,sensor_00,sensor_01,sensor_02,sensor_03,sensor_04,sensor_05,sensor_06,sensor_07,sensor_08,sensor_09,sensor_10,sensor_11
0,2018-04-01 00:00:00,2.465394,47.0920099999999995,53.2118,46.310759999999995,634.375,76.45975,13.41146,16.13136,15.567129999999999
1,2018-04-01 00:01:00,2.465394,47.0920099999999995,53.2118,46.310759999999995,634.375,76.45975,13.41146,16.13136,15.567129999999999
2,2018-04-01 00:02:00,2.444734,47.35243,53.2118,46.39757,638.8889,73.54598,13.32465,16.037329999999997,15.617770000000002,15.01012
3,2018-04-01 00:03:00,2.460474,47.0920099999999995,53.1684,46.3975677490234,628.125,76.98898,13.317420000000002,16.24711,15.6973399
4,2018-04-01 00:04:00,2.445718,47.13541,53.2118,46.3975677490234,636.4583,76.58896999999999,13.353589999999999,16.21094,15.6973399
5,2018-04-01 00:05:00,2.453588,47.0920099999999995,53.1684,46.3975677490234,637.6157,78.18568,13.41146,16.16753,15.89265,15.16204,3
6,2018-04-01 00:06:00,2.455556,47.04861,53.168399810790994,46.3975677490234,633.3333,75.81614,13.43316,16.13136,15.653929999999999
7,2018-04-01 00:07:00,2.449653,47.13541,53.168399810790994,46.3975677490234,630.6713,75.77331,13.252310000000001,16.124129999999999
8,2018-04-01 00:08:00,2.4634259999999997,47.092009999999995,53.168399810790994,46.3975677490234,631.9444,74.58915999999999,13.2884
```

- Normalising, Binning

Categorization
Sensor-0 = A
Sensor-1 = B
Sensor-2 = C
…
Sensor-51 = AZ

8-bit Linear Binning
Sensor-0 = A0 - A255
Sensor-1 = B0 - B255
Sensor-2 = C0 - C255
…
Sensor-51 = AZ0 - AZ255



Feature encoding
A0 = 2.23345/sensor-0
B23 = 47.53322/sensor-1
C143 = 53.55321/sensor-2
…
AZ41 = 4.53778/sensor-51

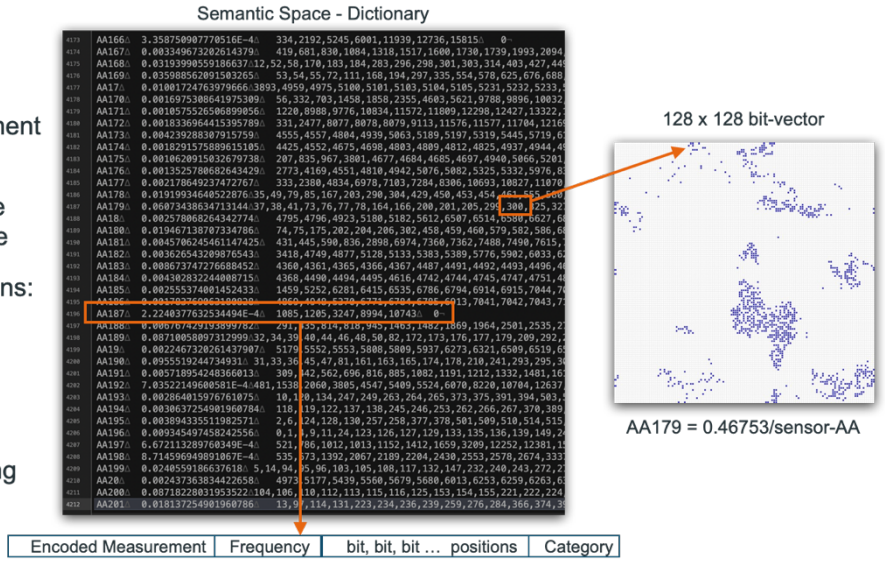
- Output Training vector

```
0  2018-04-01 00:00:00 A247 B212 C243 D226 E203 F195 G154 H175 I164 J154 K125 L202 M177 N14 O214 P1 Q160 R199 S135 T194 U227 V198
1△ 2018-04-01 00:01:00 A247 B212 C243 D226 E203 F195 G154 H175 I164 J154 K125 L202 M177 N14 O214 P1 Q160 R199 S135 T194 U227 V198
2△ 2018-04-01 00:02:00 A245 B213 C243 D227 E204 F188 G153 H174 I164 J154 K127 L205 M182 N14 O215 P1 Q160 R196 S131 T194 U228 V198
3△ 2018-04-01 00:03:00 A247 B212 C242 D227 E201 F197 G153 H176 I165 J154 K130 L207 M180 N13 O215 P1 Q160 R196 S132 T194 U227 V198
4△ 2018-04-01 00:04:00 A245 B212 C243 D227 E203 F196 G154 H176 I165 J154 K133 L209 M182 N14 O215 P1 Q160 R199 S137 T193 U228 V199
5△ 2018-04-01 00:05:00 A246 B212 C242 D227 E204 F200 G154 H175 I167 J155 K132 L210 M183 N14 O214 P1 Q160 R197 S132 T193 U227 V196
6△ 2018-04-01 00:06:00 A246 B212 C242 D227 E202 F194 G154 H175 I164 J154 K129 L211 M182 N14 O215 P1 Q161 R199 S136 T194 U228 V199
7△ 2018-04-01 00:07:00 A246 B212 C242 D227 E202 F194 G152 H175 I170 J154 K126 L210 M181 N14 O213 P1 Q160 R198 S133 T193 U225 V194
8△ 2018-04-01 00:08:00 A247 B212 C242 D227 E202 F191 G153 H175 I163 J155 K129 L214 M182 N14 O216 P1 Q160 R197 S133 T194 U228 V199
9△ 2018-04-01 00:09:00 A245 B213 C242 D227 E205 F191 G154 H176 I164 J155 K133 L215 M183 N14 O215 P1 Q160 R200 S138 T194 U228 V198
```

The pre-processing step converts raw sensor readings taken at specific timestamps into discrete vectors, each containing 50 numbered bin labels. These vectors, derived from the "NORMAL" data, are then used to train the semantic space, referred to as Retina in Semantic Folding terminology.

The vectors previously generated are inputted into the Semantic Folding Training Engine for unsupervised training, resulting in the creation of a Retina Dictionary. This dictionary comprises 2-dimensional binary vectors, or semantic fingerprints, for every measurable value from every sensor. Each binary vector is constructed with a consistent underlying topology, ensuring that each bit within the vector has a uniform meaning across all fingerprints. For instance, a '1' in bit-23 holds the same significance in any fingerprint where this bit is set.

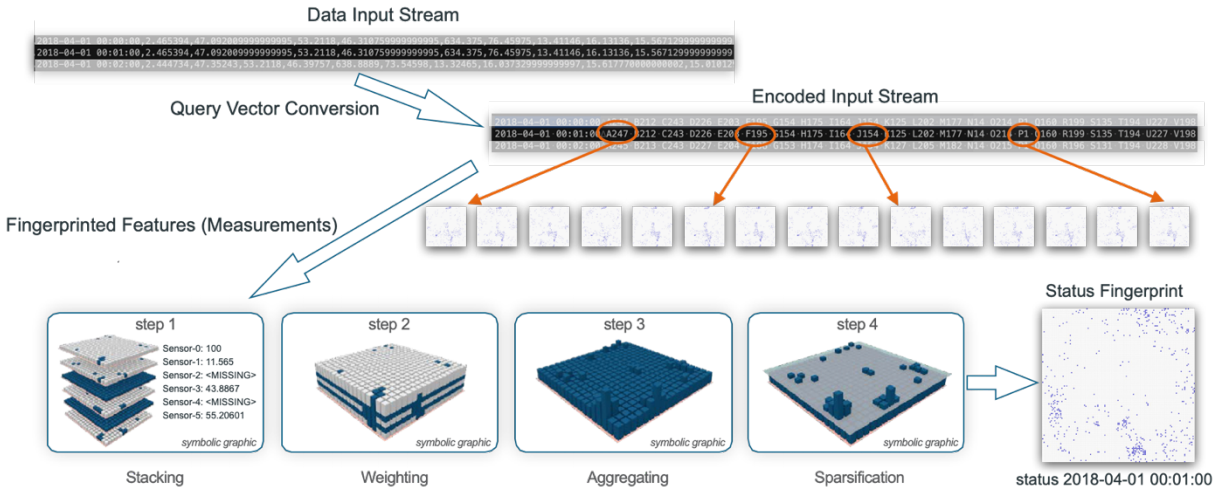
- Semantic Dictionary generation
- Every occurring measurement gets its fingerprint
- Only „Normal“ Records are part of the Semantic Space
- Semantic Space Dimensions: 8-bit/128x128
- 16384 binary features
- Training time ~ 15 min
- 7758 distinct values ranging from A1 - AZ96



An additional characteristic of these topological 2D binary vectors is their mandatory sparseness—no more than 3% of the bits are set in any given fingerprint. The Semantic Folding process has tailored the Retina for the water pump to a resolution of 128x128. Ongoing advancements in Semantic Folding technology may allow dynamic determination of fingerprint size based on the type and volume of time-series data. Current implementations typically employ 128-dimensional semantic fingerprints.

Upon concluding the water pump Retina generation, the system identified 7758 distinct measurement bins. The label of each bin, the corresponding sparse binary vector and its frequency within the training dataset is stored in the Retina dictionary.

Utilizing the trained Retina Engine, it is now feasible to perform inference by retrieving the corresponding semantic fingerprint for each measured value by finding the appropriate entry in the Retina Dictionary table. During operation, 50 measured values are recorded every minute. Each of these values is processed in the pre-processing module to generate a corresponding fingerprint. These fifty binary vectors are inherently sparse, using the fundamental property of sparse vectors: they can be combined through union (addition) without losing information.

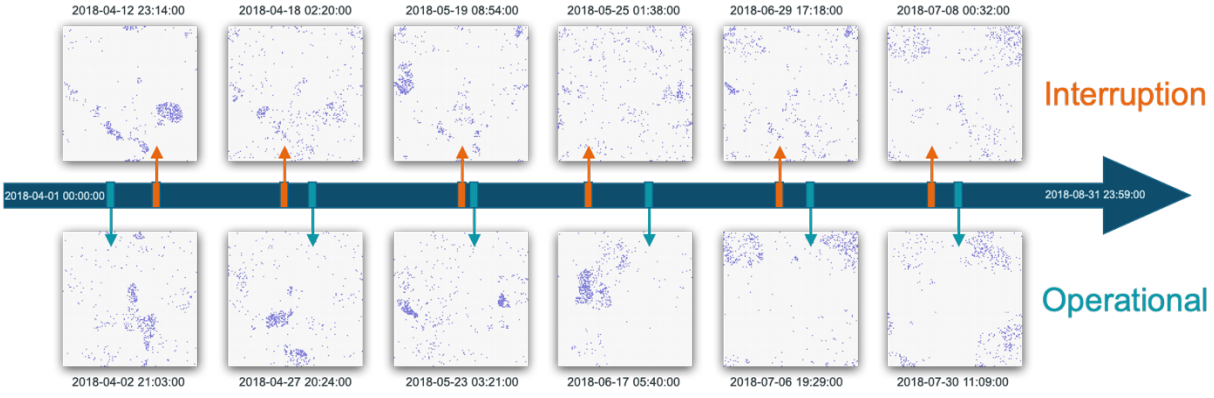


The incoming measurement fingerprints are therefore aggregated through a binary union to produce a weighted sum vector, wherein multiple bits might overlap at various positions. As more output fingerprints are added, the composite fingerprint becomes increasingly dense. To maintain the required sparsity, the sum fingerprint is re-sparsified at the end of aggregation by applying a threshold that limits the filled proportion to 3%.

Consequently, a semantic fingerprint for the entire aggregation vector is computed, covering the same semantic space as its component vectors. This composite fingerprint effectively serves as the status representation of the underlying system (the water pump), providing a comprehensive snapshot of its current operational state.

5. Results and Discussion

When the entire timespan of the pump data is converted into semantic fingerprints through the previously described method, what emerges is a dynamic chronology—a 'movie'—depicting the evolution of the

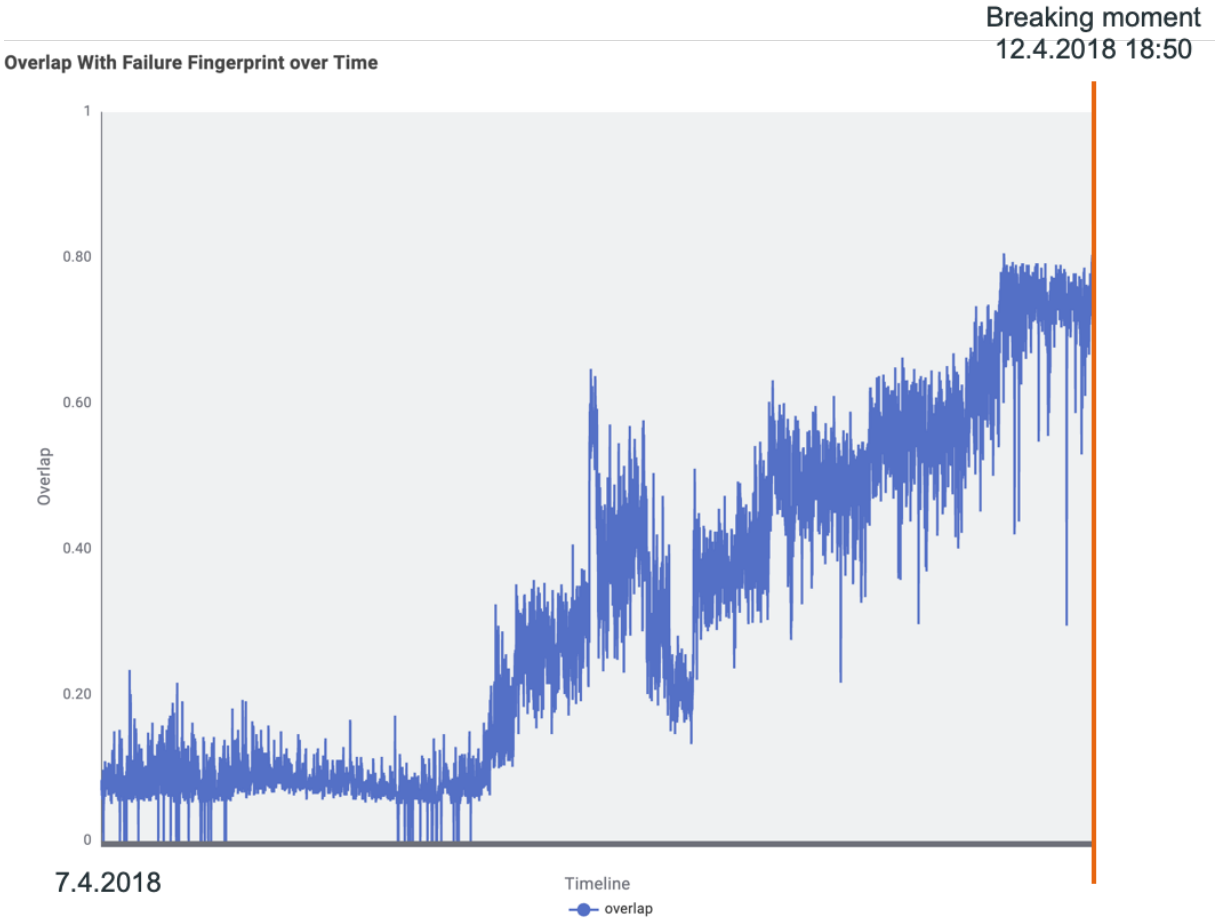


system's states. One striking observation is the high degree of resemblance between consecutive status fingerprints, as evidenced by substantial overlap values. This indicates a consistent portrayal of the pump's condition over time. As environmental variables such as ambient temperature, water pressure, or supply voltage change, the time-invariant part of the fingerprint gradually transitions into a new stable sub-pattern.

This pattern shift can be discerned in the sequence of fingerprints, even without detailed knowledge of the specific sensors involved. The fingerprint’s genuine capacity to encapsulate distinctive operational states extends to periods when the water pump is not in operation, providing a comprehensive representation of the pump's performance across all real-world contexts.
The use of semantic fingerprints to represent sensor data permits the discrimination between operational and failure states. Furthermore, it enables the subdivision of these states into specific conditions that depend on intrinsic and environmental factors.

Each status fingerprint, characterizable by specific conditions such as 'Operation after major rainfall' or 'Operation during low water levels,' can be stored and functions as a template. During operation, the live stream of fingerprints is compared to the entire template set. The degree of overlap between a current fingerprint and the templates serves as a metric that quantifies the current association with specific characteristics, for example, indicating a '63% match to low water level mode'.

Fingerprints produced during a failure state can equally serve as templates. The resulting overlap metric then provides a "predicted" measure of how closely the current state approximates a failure condition.



7. Conclusions

The Semantic Folding mechanism was applied to a dataset consisting purely of 52 sensor measurements per minute, with no additional information about the sensor types or measurement units. This dataset underwent fully unsupervised training, which was completed in just a few minutes. Subsequently, the Retina Engine was capable to convert these 52 measurements into a semantic fingerprint and effectively capture the structured information contained therein. When analyzing the temporal progression of these semantic fingerprints, a gradual change over time was observed, indicating consistency in the data representation. Notably, fingerprints recorded during normal

operation differed significantly from those obtained during pump failures. Furthermore, the predictive capabilities of these semantic fingerprints became evident, as samples closer in time to pump failures showed increased overlap with the semantic fingerprints associated with failure states, demonstrating their potential utility in predictive maintenance scenarios.

The Semantic Folding method can be applied irrespective of the complexity of the observed system, the number of sensors, or their types. It consistently produces a semantic fingerprint that represents the system's state. This fingerprint can then be analyzed to extract qualitative aspects or compared against failure states to detect anomalies in advance. In all scenarios, calculations based on semantic fingerprints are highly efficient, enabling their use in real-time and/or embedded applications. Since the method works independently of the specific system or sensors involved, it can be implemented in any setting in which a distinct system and corresponding data-sampling sources are available. Therefore, the method can be considered a foundational approach with broad applicability across various domains.

8. Future Work

Further research is required to enhance the fundamental Semantic Folding protocol by minimizing or eliminating the need for user-defined hyperparameters such as "Retina size," "binning procedure," and "sparsification."

Additionally, another research direction should focus on improving the interpretation of the resulting fingerprints, with the aim of developing parameter-free analytical tools that can effectively characterize the monitored system.

A third line of research concerns the use of semantic fingerprints as feature vectors for machine learning, enabling deep analytical or predictive solutions for highly complex systems such as organisms, integrated circuits, societies, etc.

As part of the application research, the Semantic Folding algorithm will be implemented on a universal embedded platform (System on Module, or SOM) to facilitate the flexible exploration of different instrumentation protocols in real-world scenarios.

References

1) Webber, F. D. S. (2015). Semantic Folding Theory and its Application in Semantic Fingerprinting. Retrieved from https://ar5iv.labs.arxiv.org/html/1511.08855v2
2) Pump sensor data for predictive maintenance, pump_sensor_data, https://www.kaggle.com/datasets/nphantawee/pump-sensor-data
3) “A Comprehensive Review of Multisensor Data Fusion: Foundations and Applications” by E. L. Hall and J. Llinas (1997)
4) “Decision Fusion and Feature Fusion in Pattern Recognition” by L. Xu, A. Krzyzak, and C. Y. Suen (1992)
5) "A Feature Integration Framework for the Fusion of Multi-Modal Sensor Data" by T. M. Strat and M. A. Fischler (1991)
6) "Feature-Level Fusion of Multimodal Biometric Data" by A. Ross and A. K. Jain (2004)
7) "Multisensor Fusion for Computer Vision" by J. L. Crowley (1993)
8) "A Review of Data Fusion Models and Architectures: Towards Engineering Guidelines" by Varun Khatri and James Llinas (1999)
9) "Robust Multi-Sensor Fusion for Autonomous Vehicle Navigation" by R. Toledo-Moreo and M. A. Zamora-Izquierdo (2007)
10) "Reliable Object Tracking Based on Feature Fusion" by K. Wojciechowski (2003)
11) "Feature Level Fusion for Effective Multimodal Biometrics" by H. K. Tripathi and P. G. Flikkema (2012)
12) "Sensor Fusion in Certainty Grids for Mobile Robots" by A. Elfes (1989)
13) "Fusion of Visual and Infrared Imagery for Nighttime Navigation" by A. Toet and M. A. Hogervorst (2009)
14) "Multisensor Fusion for Health and Activity Monitoring: A Review" by M. Haghi, K. Thurow, and R. Stoll (2017)
15) “A novel feature fusion network for multimodal emotion recognition from EEG and eye

movement signals." Fu B, Gu C, Fu M, Xia Y and Liu Y (2023), Front. Neurosci. 17:1234162. doi: 10.3389/fnins.2023.1234162

16) "A Survey on Sensor Fusion in Cyber Physical Systems" by Wang et al. (2018)
17) "Feature Fusion Machine: A New Multi-Modal Feature Fusion Approach" by Yang et al. (2018)
18) "Sensor Fusion for Public Space Utilization Monitoring in a Smart City" by Zanella et al. (2014)
19) "Multimodal Sensor Fusion in Automated Driving: A Survey" by Garcia et al. (2017)
20) "Adaptive Kalman Filtering for Vehicle Navigation" by Shin (2005)
21) "Efficient Multi-Sensor Fusion for Robust Autonomous Navigation" by Weiss and Brock (2011)
22) "Real-Time Sensor Fusion for Situational Awareness" by Durrant-Whyte and Bailey (2006)
23) "Feature Fusion for Image Enhancement Using Multispectral Data" by Varshney and Arora (2004)
24) "A Fast and Efficient Multi-Sensor Information Fusion System with Support of Kernel Methods" by Luo et al. (2009
25) "Efficient Sensor Fusion Techniques for Vehicle Localization and Navigation: Performance Evaluation and Reliability Analysis" by Karamat and Shafiee (2013)
26) "Scalable Fusion of Independent Component Analyses (SFICA) for Multi-sensor Data" by Zhang et al. (2014)
27) "A Scalable Approach to Multi-Sensor Integration and Fusion in Autonomous Robots" by Kim and Sukkarieh (2010)
28) "Flexible Fusion Framework for Multi-sensor Mobile Systems" by Li et al. (2015)
29) "Design of a Scalable and Flexible Sensor Fusion System for Vehicle Localization" by Chen et al. (2016)
30) Harris, Z. S. (1954). "Distributional Structure."
31) Firth, J. R. (1957). "A Synopsis of Linguistic Theory 1930-1955."
32) Rubenstein, H., & Goodenough, J. B. (1965). "Contextual correlates of synonymy."
33) Landauer, T. K., & Dumais, S. T. (1997). "A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge."
34) Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). "Efficient estimation of word representations in vector space."
35) Kanerva, P. (1988). "Sparse Distributed Memory."
36) Olshausen, B. A., & Field, D. J. (1996). "Emergence of simple-cell receptive field properties by learning a sparse code for natural images."
37) Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). "A fast learning algorithm for deep belief nets."
38) Aharon, M., Elad, M., & Bruckstein, A. (2006). "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation.
39) Makhzani, A., & Frey, B. J. (2013). "k-Sparse Autoencoders."
40) Hawkins, J., & Blakeslee, S. (2004). "On Intelligence."
41) George, D. & Hawkins, J. (2009). "Towards a Mathematical Theory of Cortical Micro-circuits."
42) Hawkins, J., Ahmad, S., & Dubinsky, D. (2011). "Hierarchical Temporal Memory including HTM Cortical Learning Algorithms."
43) Donoho, D. L. (2006). "Compressed sensing."
44) Candès, E. J., & Wakin, M. B. (2008). "An Introduction To Compressive Sampling."
45) Gilbert, A. C., Muthukrishnan, S., & Strauss, M. J. (2005). "Approximation of functions over redundant dictionaries using coherence."
46) Kutyniok, G., & Lim, W.-Q. (2013). "Compressed sensing of sparse tensors."
47) Elad, M., & Aharon, M. (2006). "Image denoising via sparse and redundant representations over learned dictionaries."
48) Luchoomun, T., Chumroo, M., Ramnarain-Seetohul, V. (2019). A Knowledge Based System for Automated Assessment of Short Structured Questions. In Proceedings of the 2019 IEEE Global Engineering Education Conference (EDUCON), Dubai, UAE, April 8-11, 2019, pp. 1349-1352.
49) Kiourtis, A., Mavrogiorgou, A., Kyriazis, D. (2020). A Semantic Similarity Evaluation for Healthcare Ontologies Matching to HL7 FHIR Resources. Studies in Health Technology and Informatics, 270, 13-17. DOI: 10.3233/SHTI200113.
50) Hole, K. J., & Ahmad, S. (2021). A thousand brains: toward biologically constrained AI. SN Applied Sciences, 3(8), Article 743. DOI: 10.1007/s42452-021-04556-5.
51) José Segarra(B), Xavier Sumba, José Ortiz, Ronald Gualán, MauricioEspinoza-Mejia, Víctor Saquicela (2019). Author-Topic Classification Based on Semantic Knowledge. In Advances in Intelligent Systems and Computing, vol. 931, pp. 132-145. Springer, Cham. DOI: 10.1007/978-3-030-21395-4_5.
52) Haj, A., Balouki, Y., & Gadi, T. (2021). Automated Identification of Semantic Similarity between Concepts of Textual Business Rules. *International Journal of Intelligent Engineering & Systems*, *14*(1).
53) Parra, E., Escobar-Avila, J., & Haiduc, S. (2018, May). Automatic tag recommendation for software development video tutorials. In *Proceedings of the 26th Conference on Program Comprehension* (pp. 222-232).

54) Herberg, J. S., Yengin, I., Satkunarajah, P., & Tan, M. (2017). Boosting Knowledge-Building with Cognitive Dialog Games. In *CogSci*.

55) Adem, K., Ilker, Y., & Dilek, K. (2018). Cognitive dialog games as cognitive assistants: Tracking and adapting knowledge and interactions in student's dialogs. *International Journal of Cognitive Research in Science, Engineering and Education*, *6*(1), 45-52.

56) Yubo, Y., Jing, M., Xiaomeng, D., Jingfen, B., & Yang, J. (2023). Data Recognition for Multi-Source Heterogeneous Experimental Detection in Cloud Edge Collaboratives. *International Journal of Information Technologies and Systems Approach (IJITSA)*, *16*(3), 1-19.

57) Dalpiaz, F., Van Der Schalk, I., Brinkkemper, S., Aydemir, F. B., & Lucassen, G. (2019). Detecting terminological ambiguity in user stories: Tool and experimentation. *Information and Software Technology*, *110*, 3-16.

58) Kamal Hossen, M., Faiad, M. A., Shahnur Azad Chowdhury, M., & Sajjatul Islam, M. (2018). Discovering Users Topic of Interest from Tweet. *arXiv e-prints*, arXiv-1803.

59) Wang, W. C., Wing, E. A., Murphy, D. L., Luber, B. M., Lisanby, S. H., Cabeza, R., & Davis, S. W. (2018). Excitatory TMS modulates memory representations. *Cognitive neuroscience*, *9*(3-4), 151-166.

60) Piumsomboon, T., Ong, G., Urban, C., Ens, B., Bai, X., & Hoermann, S. (2022, October). Ex-Cit XR: Expert-elicitation of XR Techniques for Disengaging from IVEs. In *2022 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)* (pp. 710-711). IEEE.

61) Amin, N., Yother, T., Johnson, M., & Rayz, J. (2022). Exploration of natural language processing (NLP) applications in aviation. *The Collegiate Aviation Review International*, *40*(1).

62) Van Winkle, T., Kotval-K, Z., Machemer, P., & Kotval, Z. (2022). Health and the Urban Environment: A Bibliometric Mapping of Knowledge Structure and Trends. *Sustainability*, *14*(19), 12320.

63) Hasan, H. M., & Sanyal, F. (2017). *Important keywords extraction from documents using semantic analysis* (Doctoral dissertation).

64) ALBAYRAKOĞLU, M. M., & AYDIN, M. N. (2022). INFLUENCE OF DIFFERENT THEORIES OF ETHICS ON ORGANIZATIONAL CODES OF CONDUCT OR ETHICS: A COMPARATIVE SEMANTIC ANALYSIS. *Journal of Research in Business*, *7*(IMISC2021 Special Issue), 33-47.

65) Saxena, I. (2021). *Information extraction and representation from free text reports Isha Saxena* (Master's thesis, Universidade de Évora).

66) Avioz, I., Kedar-Levy, H., Pungulescu, C., & Stolin, D. (2023). Linking asset prices to news without direct asset mentions. *Applied Economics Letters*, *30*(20), 2907-2912.

67) Pungulescu, C., & Stolin, D. Measuring Document Similarity: A Comparative Analysis of NLP Methods in Finance. *Available at SSRN 4631253*.

68) Christie, A. (2023, August). Mapping Coursework to Course Outcomes for CS Teachers Using Limited Data. In *Proceedings of the 2023 ACM Conference on International Computing Education Research-Volume 2* (pp. 95-98).

69) Kang, K., & Bae, C. (2021). Memory model for morphological semantics of visual stimuli using sparse distributed representation. *Applied Sciences*, *11*(22), 10786.

70) Pečnikar Oblak, V., Campos, M. J., Lemos, S., Rocha, M., Ljubotina, P., Poteko, K., ... & Doupona, M. (2023, August). Narrowing the Definition of Social Inclusion in Sport for People with Disabilities through a Scoping Review. In *Healthcare* (Vol. 11, No. 16, p. 2292). MDPI.

71) Wang, W. C., Brashier, N. M., Wing, E. A., Marsh, E. J., & Cabeza, R. (2018). Neural basis of goal-driven changes in knowledge activation. *European Journal of Neuroscience*, *48*(11), 3389-3396.

72) Dalpiaz, F., Van der Schalk, I., & Lucassen, G. (2018). Pinpointing ambiguity and incompleteness in requirements engineering via information visualization and NLP. In *Requirements Engineering: Foundation for Software Quality: 24th International Working Conference, REFSQ 2018, Utrecht, The Netherlands, March 19-22, 2018, Proceedings 24* (pp. 119-135). Springer International Publishing.

73) Kiourtis, A., Mavrogiorgou, A., Menychtas, A., Maglogiannis, I., & Kyriazis, D. (2019). Structurally mapping healthcare data to HL7 FHIR through ontology alignment. *Journal of medical systems*, *43*, 1-13.

74) Riikkinen, M., Saarijärvi, H., Sarlin, P., & Lähteenmäki, I. (2018). Using artificial intelligence to create value in insurance. *International Journal of Bank Marketing*, *36*(6), 1145-1168.

75) Ibriyamova, F., Kogan, S., Salganik-Shoshan, G., & Stolin, D. (2017). Using semantic fingerprinting in finance. *Applied Economics*, *49*(28), 2719-2735.

76) Pungulescu, C. (2022). Using textual analysis to diversify portfolios. *Available at SSRN 4075092*.

77) Sungur, A. K., & Surer, E. (2016, September). Voluntary behavior on cortical learning algorithm based agents. In *2016 IEEE Conference on Computational Intelligence and Games (CIG)* (pp. 1-7). IEEE.

78) Bennett, M. (2020). An attempt at a unified theory of the neocortical microcircuit in sensory cortex. *Frontiers in Neural Circuits*, *14*, 40.

79) Hawkins, J., Ahmad, S., & Cui, Y. (2017). A theory of how columns in the neocortex enable

learning the structure of the world. *Frontiers in neural circuits*, *11*, 295079.

80) Mountcastle, V. B. (1997). The columnar organization of the neocortex. *Brain: a journal of neurology*, *120*(4), 701-722.

81) Staroletov, S. (2021, May). A hierarchical temporal memory model in the sense of Hawkins. In *2021 IEEE Ural-Siberian Conference on Computational Technologies in Cognitive Science, Genomics and Biomedicine (CSGB)* (pp. 470-475). IEEE.

82) Chen, X., Wang, W., & Li, W. An Overview of Hierarchical Temporal Memory: A New Neocortex Algorithm.

83) Castro-Alamancos, M. A. (2004). Dynamics of sensory thalamocortical synaptic networks during information processing states. *Progress in neurobiology*, *74*(4), 213-247.

84) Cao, Y., Summerfield, C., Park, H., Giordano, B. L., & Kayser, C. (2019). Causal inference in the multisensory brain. *Neuron*, *102*(5), 1076-1087.

85) Nikbakht, N., Tafreshiha, A., Zoccolan, D., & Diamond, M. E. (2018). Supralinear and supramodal integration of visual and tactile signals in rats: psychophysics and neuronal mechanisms. *Neuron*, *97*(3), 626-639.

86) Jiang, X., Chevillet, M. A., Rauschecker, J. P., & Riesenhuber, M. (2018). Training humans to categorize monkey calls: auditory feature-and category-selective neural tuning changes. *Neuron*, *98*(2), 405-416.