Enquiries:

If you have questions about this document, contact openresearch@mmu.ac.uk. Please include the URL of the record in e-space. If you believe that your, or a third party's rights have been compromised through this document please see our Take Down policy (available from https://www.mmu.ac.uk/library/using-the-library/policies-and-guidelines)

A Deep Learning based Explainable Control System for Reconfigurable Networks of Edge Devices

Priyanthi M. Dassanayake[1], Ashiq Anjum[1], Ali Kashif Bashir[2,3], Joseph Bacon[4], Rabia Saleem[5] and Warren Manning[5]

[1]School of Informatics, University of Leicester, Leicester, LE1 7RH UK

[2] Department of Computing and Mathematics, Manchester Metropolitan University, Manchester, M15 6BH UK

[3]School of Electrical Engineering & Computer Science, National University of Science & Technology, Islamabad, Pakistan

[4]Make IT Different Ltd, London, EC1V 2NX, UK

[5]College of Engineering and Technology, University of Derby, Derby, DE22 1GB UK

Edge devices that operate in real-world environments are subjected to unpredictable conditions caused by environmental forces such as wind and uneven surfaces. Since most edge systems such as autonomous vehicles exhibit dynamic properties, it is clear that reinforcement learning can be a powerful tool for improving system accuracy. Successful maintenance of the position of a vehicle in such environments has been recently achieved with the aid of Deep Reinforcement Learning (DRL) that dynamically adjusts the Reconfigurable Wireless Network (RWN) response. Deep Neural Networks (DNNs) are often seen as black boxes, as neither the acquired knowledge nor the decision rationale can be explained. This raises issues with the transparency and trustworthiness of the system because the underlying AI models are not governed by any mathematical or physics laws. In this paper, we explain the process of a DNN utilisation in an autonomous dynamic positioning system by gauging reactions of the DNN to predefined constraints and capturing the associated conditions that influence the DNN in a time series. We introduce a novel digitisation technique that reduces interesting patterns of time series data into single-digits to obtain a cross comparable view of the conditions. By analysing the clusters formed on this cross comparable view, we discovered multiple intensities of environmental conditions spanning across 44% of moderate conditions and 33% and 23% of harsh and mild conditions respectively. Our analysis showed that the proposed system can provide stable responses to uncertain conditions by predicting randomness.

Index Terms—Deep Neural Networks, Explainable Artificial Intelligence, Edge Computing, Time-series data analysis, Deep Reinforcement Learning, Motif discovery.

I. INTRODUCTION

AUTONOMOUS transportation relies on Reconfigurable Wireless Networks (RWN) extensively to communicate with the destination and the source systems. Many functions in autonomous vehicles such as automatic emergency braking and lane centring assist, depend on computation, information sharing, and the timely actuation of the RWNs [1].

In real-world applications, it is inevitable for RWN enabled autonomous systems to encounter unpredictable environmental conditions, at least briefly. In operational technology, control and actuation functions in safety-critical edge systems can result in compensating actions beyond set tolerances for small fluctuations in environmental conditions. Such 'Fail-Safe' designs impact delivery, reputation, finances and ultimately, other systems [2].

The transportation industry, in particular, needs a high level of accountability, reliability and transparency. The impact of irrational behaviour can be fatal to human life, as in the 2016 Tesla Model S crash and 2018 Model X accident [3]. The machine learning algorithms such as Deep Neural Networks (DNNs) utilised in edge systems can potentially perform accurately in turbulent environments [4]. Moreover, the dynamic nature of RWN facilitates DNNs to achieve optimal or near-optimal performance. Although this significant potential suggests imminent autonomous transportation, the industry remains sceptical of utilising DNNs as the rationale behind the decisions made by the DNNs is unknown [3].

To enhance the confidence of DNN based solutions, a feature that describes the decision-making process of DNN is required. Designing and implementing such functionalities for DNN is known as eXplainable Artificial Intelligence (XAI) [5]. XAI is an exciting research field capturing the attention of a large audience both in academia and industry. Providing complete explanations of the superhuman abilities DNN have been displaying has been the main challenge that XAI faces over the years. Recently, we introduced a system displaying a remarkable ability to predict random environment conditions [4]. It consisted of a real-time, model-based application that maintains the position autonomously in an unpredictable environment by utilising model-free reinforcement learning techniques.

In this paper, the DNN response and the corresponding environmental conditions are analysed to explain the behaviour of the DNN (see Figure 1). The captured data sets, which represent time-series data, are extensive due to their velocity. They are generally analysed using sliding windows to reduce active memory space and maintenance of adequate knowledge [6]. However, these sliding windows tend to overlap, causing over counting. Once clustered, the cluster centres of over counted data behave in sine curves regardless of the shape of actual distribution [7]. Hence, for clustering, zero overlapping repetitive patterns are identified that retain the original characteristics of the datasets [8].
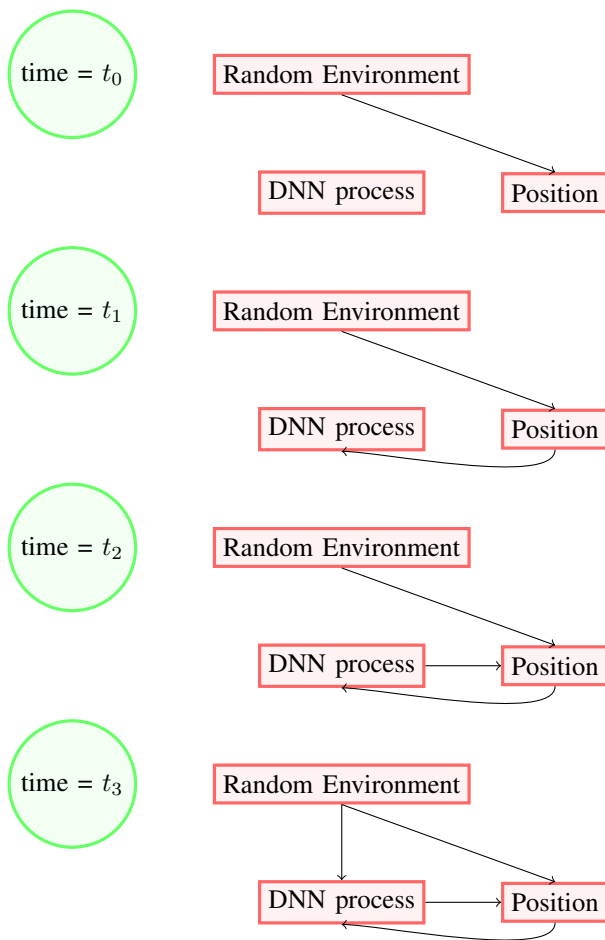
Fig. 1: Data captured in the DNN process
The environmental conditions persistently affecting the position of the vehicle not only activates the DNN but also influence the decision process of the DNN.
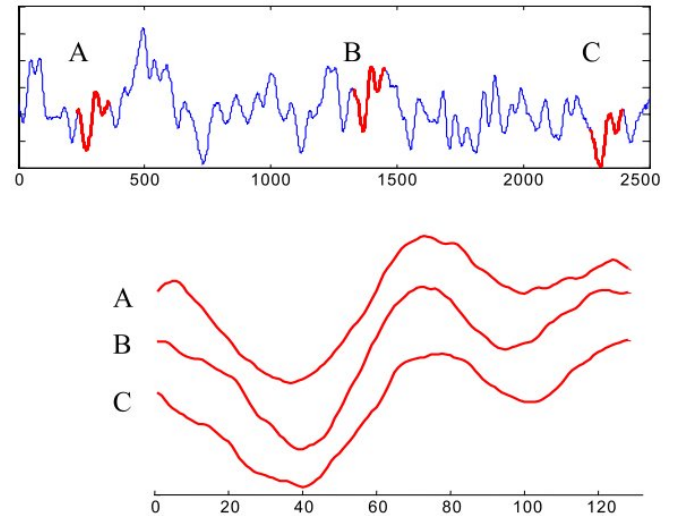

Fig. 2: A motif: 3 similar subsequences can be identified in the time series (illustration from [Lin et al., 2002])

the K-means algorithm, explain the DNN process.

To summarise, in this paper, a novel time-series data analysis model is introduced with cross comparability of various variables involved in the decision process. The paper contributes to both XAI and time-series data analysis research. The major contributions are;

1) A novel data representation technique to present the repeating patterns by a single digit which enables clustering in different dimensions
2) A novel cross comparable clustering method for time-series data
3) An explanation of the process and the acquired knowledge of DNN via cross-comparable clusters and their dispersion
4) An interpretation of the motion of the vehicle using the acquired knowledge of the DNN

The paper is organised as follows. First, Section II presents the related work conducted to describe DNN processes. Then Section III presents the data and the novel approach introduced in this paper. Section IV presents the hypothesis and the experimental environment of the approach. In Section V a discussion of the results is presented. Finally, SectionVI concludes the findings and presents future developments for the research.

Identifying repetitive patterns in time series data is known as motif discovery (see Figure 2). In motif discovery, the trivial matches around the motifs are disregarded to eliminate over counting [6]. However, this may cause losing essential information about motifs. In this paper, a novel histogram-based zero overlapping motif discovery method is introduced to include trivial matches.

The clusters of each dataset themselves, cannot explain the decision of the DNN, as it depends on the concurrent inter-related conditions. To describe DNN behaviour, the clusters of the DNN response needs to be compared against corresponding environmental conditions. The existing clustering algorithms, do not provide such functionality [9], [10].

In this paper, a novel mechanism is proposed to associate the cluster data with concurrent conditions. This is achieved by reducing the motifs into a single-digit representative of the entire motif. By applying the same process on corresponding conditions of the motif, an association is established between the DNN response and conditions encountered. When the motifs and the concurrent environmental conditions are reduced into data points, these can be plotted on a surface to generate a cross-comparable view. Clusters created on this view using

II. RELATED WORK

There have been several attempts to explain the rationale behind the black-box nature of DNN's decisions. However, many of the approaches are connected with the transparency of the logic of the algorithms or obtaining a broader perspective of the DNN's behaviour. The explainable behaviour of a DNN responding to real-world conditions has not been the primary concern of the research yet. This section, mainly presents a review of existing XAI approaches and concurrent data analysis techniques to make entry into XAI.

Coppens et. al [11] took the approach of interpreting the DNN structure to explain the process of the DNN. They

suggested mapping the DNN's classification process into binary trees of predetermined depth, where each branching node represents a hierarchical filter that influences the classification of input data to identify which input features result in specific action distributions. However, such an approach may oversimplify the behaviour of a DNN. For example, if all nodes were interpretable, then the aim for machines to outperform human processing would come to a grinding halt.

Kim and Canny [12] took a different approach to identify the behaviour of the DNN by introducing an attention model running in conjunction with the DNN which highlights the potential input features that may influence output. By identifying the actual region that the DNN considered, they concluded that the DNN causally cues on a variety of features that are used by humans while driving. Even though the authors claim that the annotation process is time-efficient for this particular case, it may not be possible to generalise for numerous other cases. Taking too long in an attempt to fully annotate a process would surely become counterproductive for many tasks.

When used in conjunction with a dynamic and transparent system, the process of a DNN becomes explainable. For example, a DNN utilised in a bio-inspired model may enable tracking the output of a DNN within set limits. Artificial homeostasis is a bio-inspired approach utilised to stabilise conditions of real-world situations. The author's previous work [4], introduced a novel bio-inspired model consisting of a hybrid of the Receptor Density Algorithm (RDA) and a Plastic Spiking Neuronal (PSN) model to emulate homeostasis by continuously placing the vehicle within the initial position and the set limit β. The RDA is an artificial immune system utilised to generate opposing force. The PSN is a threshold monitoring system inspired by the firing mechanism of neurons which is utilised to quantify the displacement from the initial position.

In unpredictable environments, generating an optimal force that emulates homeostasis requires dynamically updating the system parameters, which is not available in bio-inspired models. As a solution, Deep Reinforcement Learning (DRL) is introduced to replace the force generated by the RDA.

DRL cannot be applied in unpredictable environments due to states fluctuations caused by environmental changes. To generate reward functions the environment has to be stationary. Fortunately, the bio-inspired hybrid model encapsulated the unpredictability of the environment, enabling stable state transitions and reward functions knowable for the DRL.

The simulations showed that the DRL not only was able to generate optimal forces to withhold turbulent conditions but also kept the system within the safe limit at all times. In addition, the speed of the system surpassed the benchmark system which did not have a DNN. A closer inspection revealed that the system stayed idle for several time steps by generating optimal force according to the environmental conditions. This revealed the DNN's ability to predict the random environment.

To understand the unique behaviour of the DNN process, the output of the DNN is captured sequentially. Hence, the proposed approach described in this paper contributes to time-series data analysis research.

Due to the unique behaviour of the time series data, extracting meaningful information such as similarity is difficult compared to other data categories of data [6]. Time series analysis is considered as one of the 10 challenging problems in data mining [13]. Among them, multi-attribute time series data has been used to identify the similarity between the multiple time series data. Many of the multi-variate approaches apply distance-based methods such as lower bounding of Euclidean distance [14] and Dynamic Time Wrapping (DTW) [15]. These comparisons only show the similarity in the distribution. They do not indicate the correlation between each time series where the distributions do not match.

The motif discovery in time-series data analysis became popular after the cluster analysis using a sliding window became obsolete [8]. The clusters generated from a sliding window were found to be meaningless as the cluster centres behave like sine curves regardless of the shape of actual distribution [7]. Discovering zero overlapping motifs is necessary to obtain clusters retaining the original characteristics of the dataset.

A motif refers to a non-trivial frequent pattern. Non-trivial patterns occur visibly throughout the time series (see Figure 2). However, trivial matches can also exist few data points before and after a non-trivial match altering the starting and ending points of the motif. Including trivial matches may cause overlapping of motifs affecting the clusters to lose the original characters of the dataset.

A solution is provided by defining R radius for the non-trivial motif. A new rule is defined to ensure the motifs are at least $2R$. However, there is no explicit definition of R other than an empirical discovery per dataset. Also, the chosen size of R may affect the efficiency of the algorithm [6].

Concurrent time series data has been used for predictive analytics such as Autoregressive Integrated Moving Average (ARIMA)[21], however, concurrent time series data has not yet been utilised for motif comparison of a univariate time series. Multi-attribute time series data has been used to identify the similarity between the multiple time series data. In the realm of motif discovery in multivariate sequences, many of the applications are designed to identify motifs that matches all dimensions [10], [16], [17] or a subset of dimensions [9], [18], [20]. Many of the multi-variate approaches apply distance-based methods such as lower bounding of Euclidean distance [14] and Dynamic Time Wrapping (DTW)[15]. However, these comparisons do not provide correlations between them but only find if the distributions are similar or not.

Time-series data analysis research is one of the advanced data analysis areas. This research area requires solutions that enable the comparison of concurrent multivariate data. The proposed approach provides a novel mechanism to solve the issue of comparison of concurrent multivariate data distributions.

III. PROPOSED METHODOLOGY

This section presents, the preliminaries and summarises the notations of the use case applied for data collection. It provides the details of existing subsequence mining algorithms applied
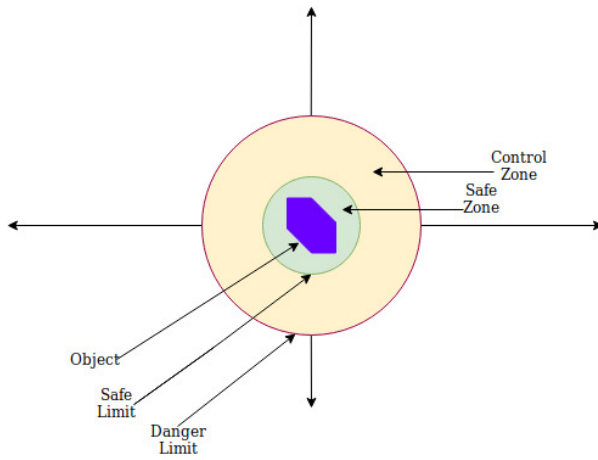
Fig. 3: Control Limits

to pre-process the data and introduces the novel subsequence representation technique to obtain a cross comparable view of subsequences.

Previous work by the authors described the use-case of the autonomous system[4]. A question of interest for autonomous transportation is maintaining the position of a vehicle in an unobservable environment. The autonomous vehicle broadly maintains its position by generating an opposing force against the resultant environment force. The objective is to maintain the position within pre-defined boundaries. Figure 3 shows control limits of the autonomous vehicle in a two-dimensional space. The objective is to maintain the autonomous vehicle position in safe zone and ensure the vehicle does not leave control zone.

The key factors to be considered are:

a) Position: The position p_t at time t indicates the deviation from the initial position. The initial position is indicated by l the maximum distance the vehicle can remain. If $p_t \ll l$ the p_t has moved further from the initial position. On the other hand if $p_t \equiv l$, this indicates that the vehicle has reached the initial position.

b) The response: The response is generated once the vehicle has been moved off the safe limit (β). It is applied towards the initial position. Once $p_t \geq \beta$ the response will be reduced using a constant decay rate until the position approached the initial position.

c) Environment: At each timestep, random data is generated from a normal distribution affecting the position. This creates an unpredictable environmental condition for the system.

This paper explores the DNN process utilised in the above use-case by investigating the influence the environmental conditions have on the decision criteria. The objective is to obtain a data model that is comparable and clusterable to explain the DNN process. The approach can be divided into three sections:

(a) Data Selection
(b) Data Representation
(c) Data Mapping

In the data selection stage, the trained state of the DNN is identified. Time-series data of variables containing envi-

ronmental conditions, corresponding positions and the DNN responses are captured for analysis. In the data representation stage, time series data are processed to obtain a comparable view of the above variables. In the data mapping stage, the relationship between obtained data representation and the DNN process is identified. This enables the explanation of the clusters derived from the above comparable view. Since the clusters are comparable with each variable and the relationship between each variable can be understood, the actions of the autonomous system become transparent.

A. Data Selection

Once a DNN has seen an adequate amount of examples, a system is assumed to be trained where no more new connections will be created. This state is identified by a predefined threshold which the error is minimised towards, known as the cost function minimisation. The cost function minimisation may not necessarily reflect a stable state as it depends on the training environment where systems may become fit for purpose before or after the system reaches the minimum cost. If the cost function is minimised before the system is ready, the consequences can be catastrophic as the untrained situations still exist in the actual real-world conditions. Conversely, if the system qualifies before it reaches the minimum cost, valuable resources are being spent unnecessarily on training an already trained system.

An obvious method to identify the stable state of a DNN is the observation of a prolonged stable state in the system. Yet, model-free DNNs do not have a mechanism to know the overall effect they have on the system's performance by applying the output. If the output can be directly compared with the system objective, the stable state of the system can be discovered.

If the position is maintained within the safe zone for an extended period, the system objective is accomplished. This point indicates that the DNN has been able to generate the exact force necessary for all the environmental conditions the system has been designed to operate. This method is system-specific, objective-oriented and can be profoundly appealing to the industry that is interested in the reliability of the output. Such an environment-specific, task-oriented solution is not only practical but also grants reliability and safety of operations. Our results showed that the data captured after identifying the stable state, the DNN displayed trained behaviour. Data about the position, the DNN response and the environmental forces are captured and saved after the identification of the stable state. These data sets comprise time-series data and represent snapshots of the system's behaviour under various conditions.

B. Data Representation

Since time is continual, time-series datasets continue to expand. To analyse these data, a method that compresses and preserves the data behaviour is required. To retain the original characteristics of the data, the approximation has to allow lower bounding [22]. Lower bounding refers to all the Euclidean distances $D(Q, S)$ between the raw data distribution

Letter	Range
a	$x_{paa} \leq -0.97$
b	$-0.97 \geq x_{paa} \leq -0.43$
c	$-0.43 \geq x_{paa} \leq 0$
d	$0 \geq x_{paa} \leq 0.43$
e	$0.43 \geq x_{paa} \leq 0.97$
f	$x_{paa} \geq 0.97$

TABLE I: Assigning Letters

Q and the compressed approximation S, there is a lower bound Euclidean distance $D_{LB}(Q,S)$ for each distribution which satisfies,

$$D_{LB}(Q,S) \leq D(Q,S) \tag{1}$$

Piecewise Aggregate Approximation (PAA) is widely used to compress data. PAA minimises dimensionality by the mean values of equal-sized frames, known as windows [14]. The window size $w = 32$ is empirically proven to be effective by previous studies [8]. Since the mean is continuous, finding matches to identify repeating mean values can be impossible. Therefore, the discretisation of these values is necessary. As discovered by [14], windows of mean values follow a Gaussian distribution and are then categorised into different ranges of Gaussian distribution annotated by characters. This process is known as Symbolic Aggregate approXimation (SAX) conversion [14]. SAX also allows the lower bounding (i.e. satisfying Equation 1) to retain the characteristics of the original data. A popular practice is defining 6 Gaussian ranges representing the characters a, b, c, d, e and f [8].

1) Preprocessing

PAA reduces the dataset of n points to n/w points by taking the average of the data items for each w sized windows. This normalised time series has a feature of behaving in a Gaussian distribution [10].

Using six ranges of Gaussian distribution (see Table I), the first six letters of the alphabet are assigned to represent each value of PAA. The dataset is discretised by translating the averaged n/w long dataset into a letter sequence. This process is known as SAX conversion [14]. At the end of the preprocessing stage letter sequences for each DNN response, position and the random environment are obtained.

If these letter sequences display repeating patterns, the rationale behind DNN's decision process can be explained. The following section describes the process of extracting repeating pattern known as motifs from these letter sequences.

2) Motif Discovery

To identify motifs, the letter sequences are initially chunked into equal parts representing, word sequences of the word length of $l = 8$. This turns the time series into a bag of words model with $(n/w)/l$ words. In our case, a word represents a timestamp of $8x32 \approx 250ms$. If there are similar repeating patterns within the above timestamp, unique repeating words are observable. In this bag of words, these unique words and their repeating frequencies can be identified by plotting them on to a histogram.

Figure 4 illustrates the frequency against the histogram of unique words. The highest number of word count indicates the most frequent repeating words. If there are unique words with the same frequency, they may be adjacent to each other and
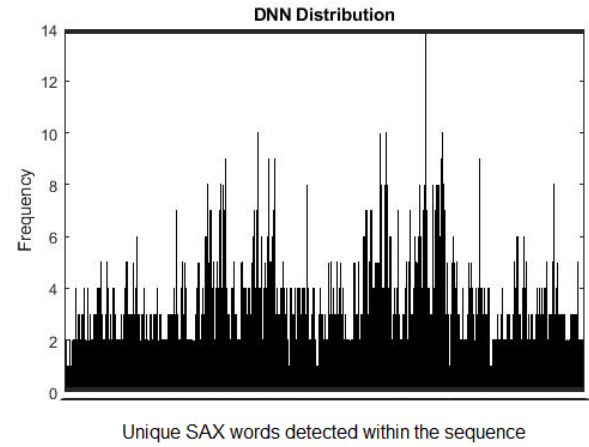

Fig. 4: Motif Distribution
The unique eight-letter long SAX words with their frequencies distinguishing the frequent repeating patterns.
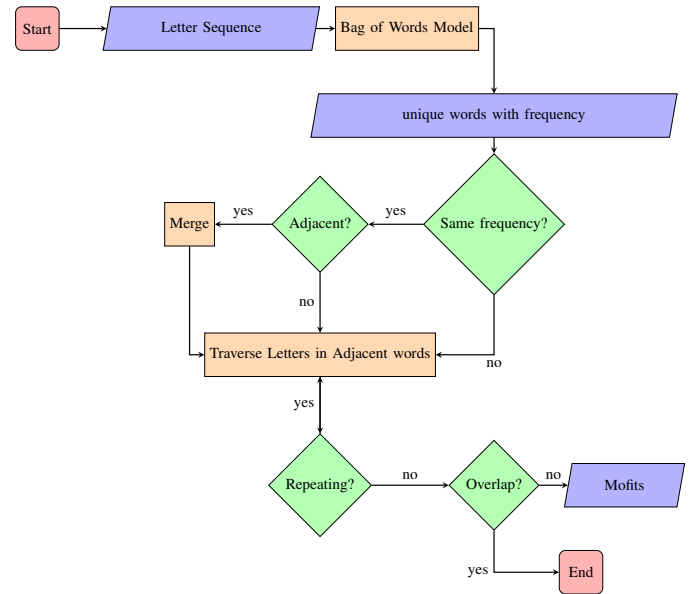

Fig. 5: Motif Discovery

are also repeating. By checking the indices of the words with the same frequency, they can be merged to form a ϕl long motif where ϕ refers to the number of unique words with the same frequency which are next to each other.

Also, the adjacent words to the repeating pattern may contain repeating letters which are $< l$. By traversing on to the adjacent words for repeating letter sequences, the motif can be expanded further. Hence, with the bag of words model, the maximum length m_d of a motif m can be found as

$$m_d = \phi l + (l - j) + (l - k) \tag{2}$$

where $0 < j, k \leq l$ refers to the number of non-repeating characters of the adjacent words. Figure 5 summarises the above process.

	a	b	c	d	e	f
a	a to a	a to b	a to c	a to d	a to e	a to f
b	b to a	b to b	b to c	b to d	b to e	b to f
c	c to a	c to b	c to c	c to d	c to e	c to f
d	d to a	d to b	d to c	d to d	d to e	d to f
e	e to a	e to b	e to c	e to d	e to e	e to f
f	f to a	f to b	f to c	f to d	f to e	f to f

Fig. 6: Transition Matrix

(a) DNN response   (b) Position   (c) Environment

Fig. 7: BitMap images

(a) Environment 1   (b) Environment 2   (c) Environment 3

(d) DNN response 1   (e) DNN response 2   (f) DNN response 3

(g) Position 1   (h) Position 2   (i) Position 3

Fig. 8: DNN response for a random environment with a constant resultant force of 5.6 units
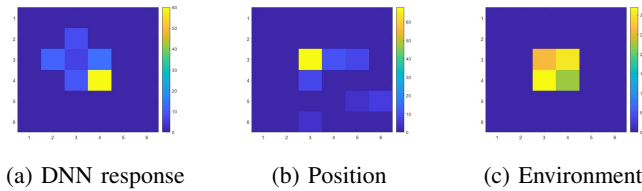
It is assumed that the frequent patterns of interest have a minimum of l letters. This is a more straight forward definition to eliminate overlapping motifs than defining a non-generic distance of $2R$ [8]. Overlaps can be eliminated by ignoring the letters with the same index. Hence, this novel method guarantees that the motifs are not overlapping with each other. Any clustering performed on these motifs will generate meaningful results since the motifs have the structure of zero-overlaps.

3) Visualisation

To understand the behaviour of motifs, they have to be visualied, along with their transitions. For example, if an identified motif consists of the "$abcfefaea$" sequence, 8 transitions can be identified as [a to b], [b to c], [c to f], [f to e], [e to f], [f to a], [a to e] and [e to a]. The frequency of these transitions can be captured into a transition matrix (see Figure 6).

Since there are 6 letters, the motifs are captured into a 6x6 matrix where each item on the matrix represents the frequency of the corresponding transition that occurred in the motif. Using the indices of the motif, the information parallel to the motif regarding the environment and the position can be captured into a transition matrix. By assigning colour codes for the frequency of transition, the matrix can be visualised as a bitmap image. Figure 7a displays a motif and its transition frequencies. The indices of the motif can be used to identify the position information and environment conditions. Figure 7b and Figure 7c depict the corresponding bitmaps for the position and the environment. This visualisation is inspired by Kumar et al.'s [24] bitmap generation technique which counted two-letter sequences for the bitmap. Rather than combining two-letter sequences, a transition matrix that counts transitions from each letter is used to describe the motifs. This method allows retaining all the information about actual transitions, leading to the discovery of acquired knowledge of the DNN.

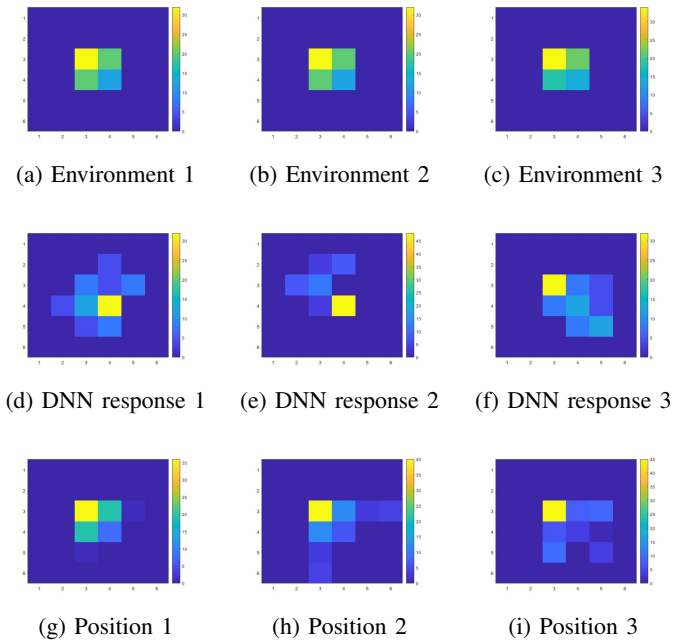In the majority of bitmaps, the position uniquely remained at position c displaying the highest frequency on the bitmap image. The position c signifies the range of -0.43 to 0 in the Gaussian distribution. The data distribution displaying a higher probability near zero and under indicates that in the majority of the time, the position remained below its mean value. This indicates that in almost all instances, the DNN has been able to bring the position towards the initial position. In contrast, many of the bitmaps representing the environment displayed higher frequencies within the range of -0.43 to 0.43. This exemplifies that generating random environment forces from a Gaussian distribution causes the data to behave in a Gaussian manner.

To compare the patterns, similar motifs can be categorised and the corresponding behaviour of the environment and position can be observed. In contrary to our expectations, similar motifs displayed neither a similar environment nor position bitmaps. Also, for a unique environmental force, the bitmap images of motifs did not display a unique behaviour (see Figure 8).

This is because the resulting condition is affected by not only just the highest transitions but also the overall transitions caused by less frequent transitions. For example, according to Figure 7c, the environment forces have random motion dominated by 'c to d' and 'd to c' motions. Since this motion cancels out each other, the random environment has been dominated by forces in the region of c. Hence, the visualisation itself would not provide a complete picture of the situation when forces cancel each other out. Hence, we introduce a novel drift based technique to reduce the bitmap images into a single digit.

Current Position	$n_x > 0$	$n_x < 0$
a	–	$-0.97 - n_a$
b	$-0.97 - n_b$	$-0.43 - n_b$
c	$-0.43 - n_c$	$0 - n_c$
d	$0 - n_d$	$0.43 - n_d$
e	$0.43 - n_e$	$0.97 - n_e$
f	$0.97 - n_f$	–

TABLE II: Drift Calculation

4) Digitisation

To convert a Bitmap into a single digit, the positions of the transition matrix are investigated. The diagonal of the transition matrix indicates the transition on to the same position, i.e. stable states. The transition from a to b is not similar to the transition from a to f. To emphasis this, each transition is increased by moving away from the diagonal values by α, where $0 < \alpha << 1$. This implies that the motion from a to f is $5 \times \alpha$ more intense than no motion, whereas, the transition a to b is only $1 \times \alpha$ stronger than no motion. Moreover, in each row of the matrix, up to the diagonal value, the transitions indicate motion towards the left end of the Gaussian distribution. The values after the diagonal value indicate motion in the opposite direction. To illustrate this phenomenon, diagonal values are annotated and the values under the diagonal identified as positive and the values above the diagonal as negative. If these values are summed up values in row-level, the bitmap of the 6x6 matrix can be reduced to a 6x1 vector. Similarly, a column-level summation will produce a 1x6 vector. Therefore, by applying the directions and the intensities of the forces, the transition matrix can be reduced into meaningful rows and column vectors.

These vectors display the accumulated drift that occurred on each letter range from its assigned range. For example, the existence of ones in a row across the transition matrix at the letter d may result in the row sum to be $1(1 + 3 \cdot \alpha) + 1(1 + 2 \cdot \alpha) + 1(1 + 1 \cdot \alpha) + 1 - 1(1 + 1 \cdot \alpha) - 1(1 + 2 \cdot \alpha) = 2 + 3 \cdot \alpha$. This represents that the vehicle has been moved towards the initial position. To position this drift $(2 + 3 \cdot \alpha)$ within the range $[0, 0.43]$, it has to be transformed into a z-normalised value n_d using the overall mean and standard deviation of all drifts in the column vector. If this value is positive, the letter d's stating position can be drifted towards the letter c. This can be represented by reducing the n_d value from the starting position 0 which may land on a position of a, b or c. Similarly, if the value n_d is negative, the starting position of the d can be replaced by reducing it from the ending position 0 resulting in a new position of e or f. A lookup table for the drift calculation for the SAX transformation of this example can be derived as in Table II). Once the accumulated drifts are mapped onto the SAX ranges with the aid of z-normalisation, new SAX representations may occur replacing the existing letter represented by a particular position.

The drift from each letter can be assigned as follows. Let a SAX letter x represents a range of values starting from p_x and ending at q_x of a Gaussian distribution. If n_x refers to the z-normalised value of the collective drift occurred in a summed up vector either in row level or column level, The
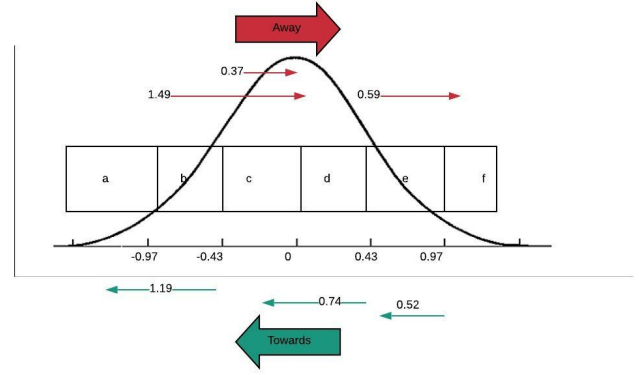

Fig. 9: Drift from the SAX letters

drift occurred from the SAX letter d_x, can be identified as:

$$((n_x > 0) \rightarrow d_x = (p_x - n_x))$$
$$\wedge ((n_x < 0) \rightarrow d_x = (q_x - n_x))$$
$$\wedge ((n_x = 0) \rightarrow d_x = 0)) \quad (3)$$

These drifted vectors themselves would not provide a concise picture of the resultant force. In the row vector, values at the first row tend to have a negative value and the values at the sixth row can never get a negative value. Similarly, when the sum of the column vector of 1x6 elements is considered, vice versa is true. Thus, the intensities between these two vectors have to be identified.

The dot product represents the directional multiplication. It is an indication of intensity between two vectors due to their alignment. Assume the row vector is denoted by $\mathbf{X}$ and the column vector is denoted by $\mathbf{Y}$. The dot product is represented by;

$$\mathbf{X} \cdot \mathbf{Y} = |X||Y|cos\theta \quad (4)$$

According to Equation 4, the dot product of row and column vectors essentially represents the resultant force of the two vectors. Hence, the dot product of the z-normalised row and column drift vectors provide a concise and meaningful representation of the entire transition matrix.

The following example explains the digitisation process applied in a matrix. Consider the below matrix. Assume that, each item in the matrix represents the frequency of transitions that occurred in a motif represented by the position of the transition matrix. For example, 2 represented at the position $(1, 5)$ indicates that there were two transitions of $[a$ to $e]$ that occurred in this particular motif. The matrix on the right is derived by applying the motion toward the initial position as positive and motion away from the initial position as negative. The non-diagonal values are then increased by $\alpha = 0.01$, ranging from 0.01 to 0.05 depending on the intensity of the transition.

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 2 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 3 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \rightarrow$$

$$\begin{bmatrix} 0 & 0(1.01) & -1(1.02) & -2(1.03) & -2(1.04) & -1(1.05) \\ 0(1.01) & 0 & 0(1.01) & -1(1.02) & 0(1.03) & 0(1.04) \\ 3(1.02) & 3(1.01) & 0 & 0(1.01) & 0(1.02) & 0(1.03) \\ 0(1.03) & 0(1.02) & 0(1.01) & 0 & -1(1.01) & -1(1.02) \\ 1(1.04) & 1(1.03) & 1(1.02) & 1(1.01) & 0 & 0(1.01) \\ 0(1.05) & 1(1.04) & 1(1.03) & 1(1.02) & 0(1.01) & 0 \end{bmatrix}$$

By adding up each row, the 6x1 row vector X is derived from the above matrix. This vector can further be normalised to compare the drift from the Gaussian range of the SAX letters using z-normalisation.

$$X = \begin{bmatrix} -5.18 \\ -1.02 \\ 6.09 \\ -2.03 \\ 4.10 \\ 3.09 \end{bmatrix} \xrightarrow[Z-normalisation]{((X-\mu)/\sigma)} \begin{bmatrix} a \xrightarrow{[-1.4100]} b \\ b \xrightarrow{[-0.4359]} c \\ c \xrightarrow{[+1.2289]} b \\ d \xrightarrow{[-0.6724]} e \\ e \xrightarrow{[+0.7630]} d \\ f \xrightarrow{+0.5265} e \end{bmatrix}$$

These values enable capturing the actual drift from the SAX labels. Figure 9 depicts the drifts from each SAX letter. The values of the 6x1 vector indicate whether the vehicle moves towards the initial position or away from the initial position. For example, a negative value of the first item ($a = -1.41$) indicates that the vehicle is moving towards b whereas a positive value on the fifth item ($e = 0.76$) indicates that the vehicle is moving towards d.

Therefore, the resultant matrix can be identified as;

$$X = \begin{bmatrix} -0.97 - (-1.41) \\ -0.43 - (-0.4359) \\ -0.43 - 1.22 \\ 0.43 - (-0.67) \\ 0.43 - 0.76 \\ 0.97 - 0.52 \end{bmatrix} = \begin{bmatrix} 0.4400 \\ 0.00591 \\ -1.6589 \\ 1.1024 \\ -0.3330 \\ 0.4435 \end{bmatrix}$$

Similarly, by applying the same normalisation and drift calculation process, the column vector for the above example can be derived as,

$$Y = \begin{bmatrix} 0.0036 & -2.2723 & -0.4876 & 0.6935 & 2.17 & 1.89 \end{bmatrix}$$

Accordingly, for the above example, the bit map can be represented by the dot product of X and Y.

$$\mathbf{X} \cdot \mathbf{Y} = 1.6767$$

C. Data Mapping

It is necessary to understand whether the values calculated by the digitisation process accurately represent the conditions of a bitmap. To analyse this, bitmap images having the same digitised value are captured. Figure 8a,8b and 8c illustrate the environment bitmap images having the same digitised value at different positions. The almost identical nature of these bitmaps confirms that the digitisation process provides an accurate picture of the overall transitions of a bitmap image.

To understand the relationship between the digitised values and the DNN process, experiments were undertaken with 36 matrices initialised with a single value on each transition and setting all the other transitions to zero. Table III displays the values derived from the digitisation process. Figure 10

provides a visual representation of these values. According to Figure 10, the positions that affect the overall digitised value positively are stable positions which are indicated by the diagonal values. The end cases of the diagonal values remain small. These are expected behaviours, as for the end cases, comparisons against one another are not possible. All the other positions in the transition matrix do not indicate a drastic increase. However, when the stability is broken, the values become distinctively small. These accommodate the detection of unstable conditions.

The digitisation process, which represents numerical values for an entire phenomenon provides an accurate picture of the underlying situation of the DNN with minimum information. Compared to curve fitting and graph analysis, digitisation enables the condensing of the data. As a result, the process enables the extraction of higher levels of knowledge from a particular dataset as well as gaining a broader perspective of the underlying process. In the following sections, we describe the relationship between the digitised value sand the DNN process.

1) The Initial Position

A row-wise comparison with the transition matrix indicates that the highest digitised value in each row is found at the diagonal. Diagonal values represent stable states, where there is no motion. As soon as this stability is broken, the minimum value of each row can be seen. The transition from b to c, next to the stable state b and the transition from e to d, next to the stable state e display the overall minimum value. These values happened to lie in columns of c and d, where the mean value, zero of the Gaussian distribution is also found. Hence, the initial position of the vehicle lies between position c and d.

2) The Safe Limit

As described above, the vehicle's motion away from the initial position is indicated by an abrupt minimum value. However, columns b and e, except when positioned next to the diagonal, display higher values than the adjacent values. This rise indicates another stable state. This can be mapped to indicate two ends of the safe limit lying in the transitions from other positions to b and e.

3) The Danger Limit

The diagonal values at a and f are relatively smaller than the other diagonal values which can be mapped on to the end cases. It is interesting to find that the position a, the position a to f and the position f to a has the same digitised value. This implies that moving the vehicle to a or f has the same impact. This indicates that both a and f represent two ends of the danger limits.

4) The Impact

Two conditions can be derived from the digitised values, stability and motion. As described above, the diagonal values, safe limits and the danger limit display larger digitised values. These represent defining stable states of the vehicle. However, according to Figure 10a, as these stable states move towards the danger limit, the values decline linearly. Figure 10b display that the values are identical in the negative and the positive halves of the motion. For example, if Figure 10b is folded through the diagonal of the top left corner and either part is rotated 180^0, they look identical. It is noticeable that the
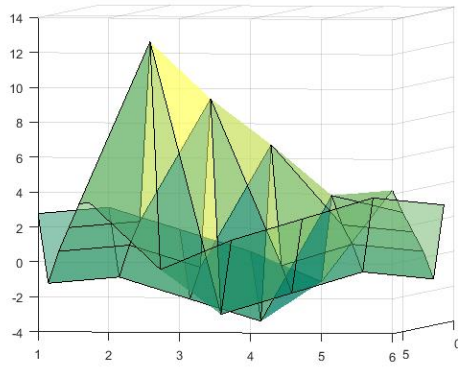
(a) Impact of individual positions which may affect on overall decision of digitisation
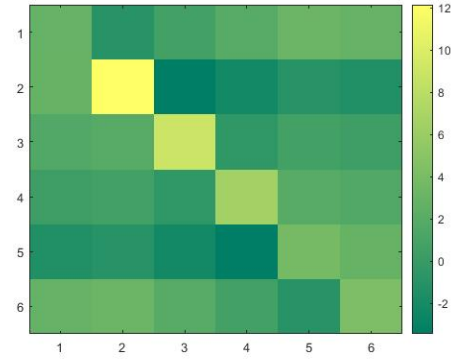
(b) Contribution of individual positions

Fig. 10: The experiment of the transitions matrix

	a	b	c	d	e	f
a	2.7694	-1.0353	0.67125	1.9094	3.15	2.7694
b	2.8343	12.1520	-3.4491	-2.2108	-0.9702	-1.3508
c	1.5930	1.9744	9.025	-0.504	0.736	0.3556
d	0.3556	0.7362	-0.5043	6.5489	1.974	1.593
e	-1.3509	-0.9703	-2.2109	-3.4491	3.7815	2.8344
f	2.7694	3.1500	1.9094	0.6712	-1.0353	4.2325

TABLE III: Impact of Digitisation

parallel diagonal values in these halves also follow the same linear decline. This concludes that motion is represented by smaller values. A digitised value can be mapped onto the stability of the vehicle, if it contains larger values and it can be mapped onto the motion of the vehicle if it contains smaller values.

The overall value of a bitmap obtained from the variables in the dynamic positioning can be mapped as follows. Obtaining a larger digitised value on a position bitmap indicates that the position is stabilised, while a smaller value indicates an unstable position. On the contrary, a smaller value obtained from a DNN response indicates an abrupt motion, which implies applying higher opposing force. Similarly, a larger value indicates a milder opposing force. This motion factor can be applied to environmental conditions as well. For example, a larger value in the random environment may indicate a milder condition while a smaller value may indicate a rough condition. This indicates that the overall value of the bitmap can be considered as an indication of either the vehicles stability, the intensity of opposing force or the severity of the environmental conditions.

IV. EXPERIMENTAL ENVIRONMENT

In this section, the experiment set up for this time-series data analysis problem is presented. Since the DNN reacts according to the environmental conditions, the environmental conditions serve as the independent variable of this experiment. The environmental conditions are derived from a normal distribution. The reaction to random environmental conditions captured as the DNN response is considered as the dependent variable.

The conditions such as the initial position, the safe limit, the danger limit and the maximum force allowed by the DNN are maintained constant throughout the experiment. These constraints are observed in the position. While maintaining the control limits constant, the experiment captures various responses of the DNN for a range of random environmental conditions obtained from a normal distribution

The experiment hypothesises that the DNN can dynamically adjust its response to maintain the position of the vehicle within constraints under various random environmental conditions. To test this hypothesis, first, motifs of DNN response and the corresponding environmental conditions are detected. By reducing the environmental conditions into single digits using the digitisation technique described above, the DNN response is categorised under these unique values. The experiments found 1,809 positive and 434 negative unique environmental conditions. A collection of 5,110 DNN responses are categorised under these unique values. These DNN responses are digitised to obtain a Cartesian plane of environmental conditions representing the x-axis and DNN responses representing the y-axis. Once the positions corresponding to each DNN response is digitised, they can be introduced as the z-axis. As a result, a 3-dimensional view is obtained, containing environmental conditions, DNN responses and positions. By performing clusters on this comparable view, meaningful information is derived to test the above hypothesis.

The experimental platform consisted of two Intel(R) Core(TM)2 Quad 2.5 GHz CPUs machine running on Windows 10. The data captured from the bio-inspired system is stored in 3 text files of 83 MB containing random environmental conditions, DNN responses and the position data. The experiments were implemented on MATLAB R2019b environment. The algorithm generates over 5000 bitmaps and these are used to calculate the digitised values. These values generate several graphs and charts to test the above hypothesis. The Github repository containing the MATLAB codes of the experiment can be found at rb.gy/8dd7l7.

(a) DNN response vs Random Environment   (b) Position vs Random Environment   (c) DNN response vs Position
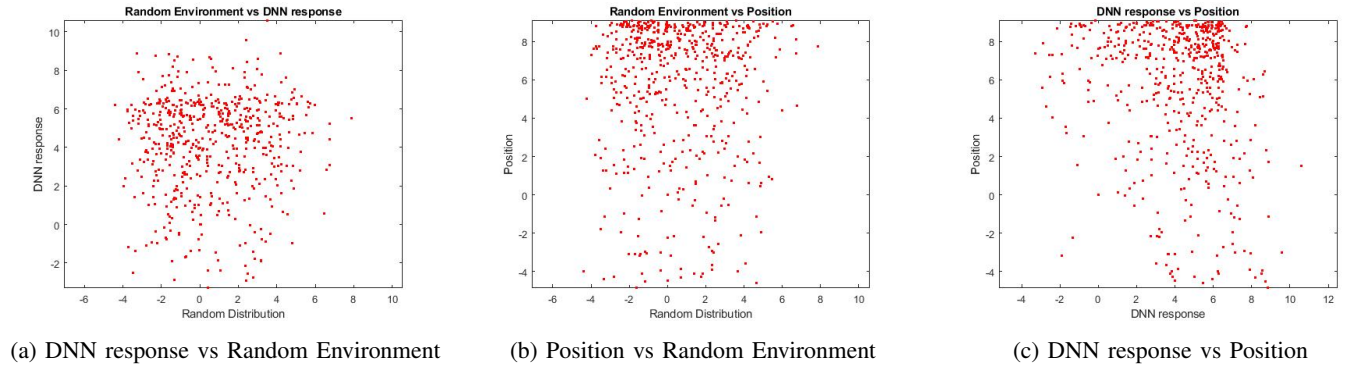
Fig. 11: Pairwise comparison of variables

V. RESULTS AND DISCUSSION

The results generated by the digitised motifs of DNN responses, the corresponding environmental conditions and the corresponding positions the vehicle remained under those conditions are plotted onto scatter diagrams. This enables both the correlations and visualising clusters to be captured. This includes two stages.

1) **pair-wise Comparison:** The relationships between each variable are identified using pair-wise comparison and scatter diagrams reveals correlations within the variables.

2) **Clustering:** The actual distribution of the motifs and the corresponding environmental conditions are visualised by representing these variables as data points in a 3-dimensional space. These data points are then clustered using the K-means algorithm to explain the decision.

a) pair-wise Comparison: Figure 11 shows the scatter diagrams of 3 pairs of DNN responses, environmental conditions and positions. The relationship between the DNN response and random environmental conditions is shown in Figure 11a. Although no correlation is visible, this graph displays that the environmental condition has received various DNN responses. This is because the DNN not only responds to the environmental conditions, but also to the position to achieve the system objective. Figure 11b displays the relationship between the position and the random environment. Notice that the position indicates a higher value the majority of times. This shows that the position has been stable on many occasions. The skewness to the left of the concentrated points indicates that the was position managed stable operation under turbulent environment conditions (recall that smaller values indicate motion). Figure 11c represents the relationship between DNN responses and the position. The skewness to the right of the concentrated points indicates that DNN did not have to abruptly respond to the position. These graphs claim that the DNN has been able to maintain a stable position, under most environmental conditions without much effort.

While the above comparisons do not justify the hypothesis as the dynamic aspect of the DNN could not be understood completely, it seems that the DNN effortlessly maintains the position stable in turbulent conditions, which is a behaviour not normally excepted by a human operator. Therefore, to understand the fundamentals of the DNN, further analyses is needed. The decision rationale of the DNN has to be fully understood to explain this behaviour.

b) Clustering: To cluster data, a 3-dimensional view of DNN responses and corresponding random environmental conditions and position is formed. Figure 12 exhibits the application of the K-means algorithm on the digitised data points under various clusters. Unlike Figure 12a and Figure 12c, the digitised dimensional representation of Figure 12b exhibits three distinct clusters. Therefore, for our analysis, the digitised motifs are clustered into three clusters.

$$Centroids \equiv \begin{bmatrix} & C1 & C2 & C3 \\ Environment & -1.23 & 2.71 & 0.53 \\ Position & 7.75 & 7.54 & 0.75 \\ DNN & 4.12 & 5.12 & 5.02 \end{bmatrix}$$

The centroid values, which are the central points for the distribution of each three clusters display three distinct behaviours of the DNN process. Clusters $C1, C2$ and $C3$ consist of $43.58\%, 32.86\%$ and 23.55% of digitised motifs respectively. Cluster $C1$ has the tightest spread of data of an average 2.77 ± 1.3 distance from the centroid. Clusters $C2$ and $C3$ have a spread of 2.84 ± 1.7 and 4.08 ± 1.6 respectively. In the next section, these centroids and their spread is analysed to understand the decision rationale and the acquired knowledge of the DNN.

A. Decision Rationale

Various responses to environmental conditions from DNN and the position can be found in the centroids $C1, C2$ and $C3$. The moderate range of DNN responses within $4.2 - 5.15$ indicates that the DNN responded stably to all environmental conditions. The environmental conditions, however, have been constantly volatile compared to the position and the DNN response within the range of $-1.22 - 2.78$. The position remains relatively stable in rapid environmental changes and relatively slower environmental changes having relatively large values over 7.6. In relatively moderate environmental conditions, the position has been in constant motion. To understand the responses clearly, the clusters are divided into 3 categories
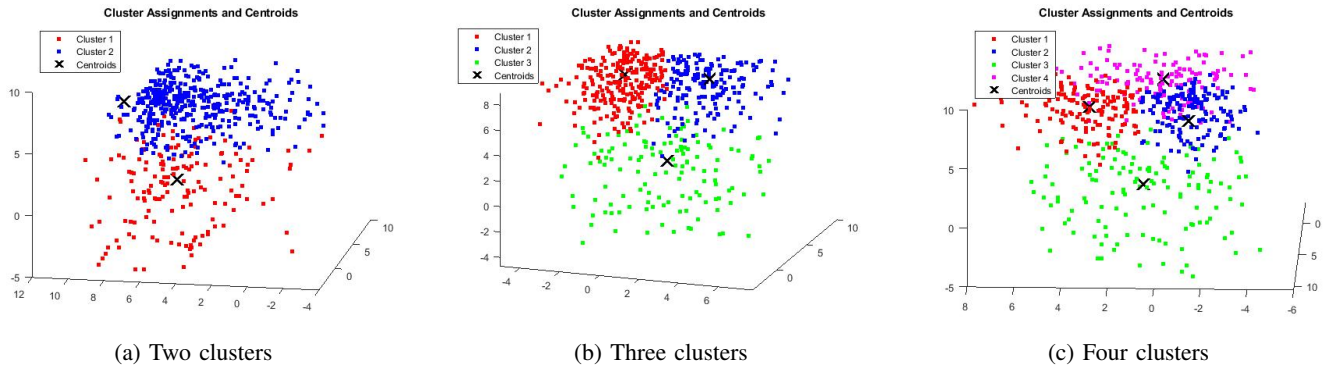
(a) Two clusters   (b) Three clusters   (c) Four clusters

Fig. 12: Clusters in 3 dimensional view

according to environmental conditions, harsh (the smallest value -1.22), mild (the largest value 2.78) and moderate.

1) Harsh Environment-$C1$

In this situation, the DNN applies a relatively large force. It is fascinating to discover that under harsh environmental conditions a position remained most stable. This cluster has the tightest spread (2.76 ± 1.3). This indicates that the intensity of environmental forces did not vary and the DNN did not have to use a wide range of forces for rapid environmental changes. To identify the position we compared the range $7.75 \pm spread$ against the values of Table III. We found that the calculated values were equivalent to the values obtained while the position remained closer to the initial position. By applying forces in almost similar intensities the DNN had not only been able to tame harsh environmental conditions but also maintain the system in the most stable state.

2) Mild Environment- $C2$

As the environmental changes start to slow down, the DNN applies relatively smaller forces. The position remains stable without much effort. Similar to $C1$, $C2$ is dispersed in a relatively tight area (2.84 ± 1.7). This indicates that the DNN uses similar strategies for a tightly distributed range of mild conditions. Similar to the harsh conditions, the dispersion also indicates that the position remained closer to the initial position when compared to the range of $7.54\pm spread$ against the values of Table III. It is interesting to see that the DNN is in a relaxed state compared to the harsh conditions yet able to achieve almost similar results.

3) Moderate Environment-$C3$

Even though relative DNN response displays a moderately stable response of 5.02 units, the cluster has the largest spread (4.06 ± 1.6). The larger range of dispersion indicates that the intensities of the environmental conditions and the corresponding strategies of DNN have a higher variance compared to the other instances. The position of value 0.75 indicates that in moderate conditions the positions fluctuates rapidly. However, when the range $0.75 \pm spread$ is compared against Table III, it lies between the safe limits. This implies that in moderate conditions, regardless of moderate response, the DNN applies various intensities to maintain the vehicle within the safe zone.

The centroid and dispersion analysis confirm that DNN is using multiple strategies to maintain the vehicle within the
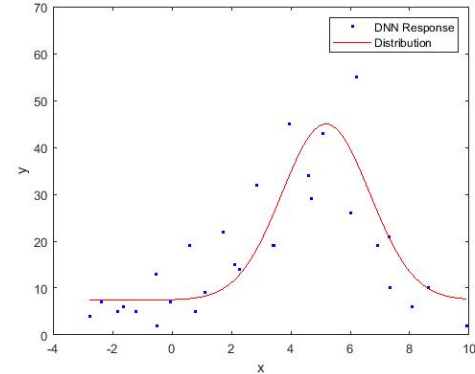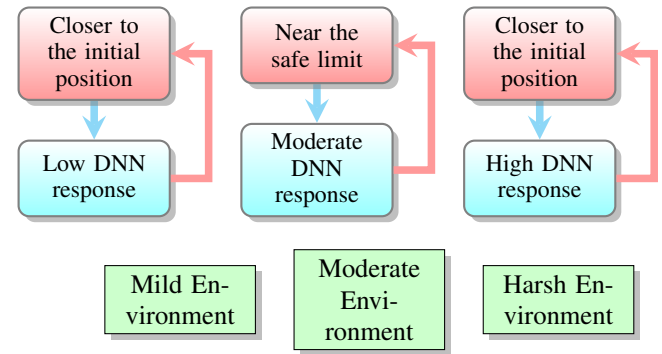

Fig. 13: Strategies of DNN

safe limits (see Figure 13). It also implies that the DNN was able to detect the environmental conditions and stably react to them. Understanding the basis of this decision process remains a question. The next section, discusses the acquired knowledge of the DNN which led to choosing the right response.

B. Acquired Knowledge

In the training state, the DNN learns and acquires knowledge about the system, which will be applied in the testing state. Traditional methods of evaluating weights, cannot be used to explain this implicit knowledge. In the previous section, the clusters and their dispersion were scaled on to the motion of the vehicle. In this section, they are used to extract implicit knowledge and explain the process of the DNN.

1) Data Distribution

Did the DNN already know from which distribution the data was coming from? If so was it able to predict the future conditions? To answer these questions the number of data points categorised into harsh(44%) mild (23%) and moderate (33%) environmental conditions are considered.

According to the probability of environmental conditions, nearly half of the points (0.44) are concentrated within the range $(-1.22 \pm spread)$. The mean of these values is closer to zero. Similarly, the means of the mild $(2.71 \pm spread)$ and moderate $(0.53 \pm spread)$ conditions can be approximated to positive values less than 3. If these points are arranged in a number line in ascending order and plotted against the probability density, the following conditions can be observed.

Mild conditions may display a growth exponential function (e^x) starting from zero until it reaches the peak at the left end of the harsh conditions which are concentrated around zero. Similarly, the moderate environmental conditions may display a decay exponential function $(\frac{1}{e}^x)$ starting from the right end of the harsh conditions and decaying to zero. This is similar to a bell-curved function. Equation 5 mathematically represents a Gaussian distribution. Once the clusters are arranged according to the identified environmental conditions, the resulting distribution can be approximated to a Gaussian distribution.

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \tag{5}$$

This shows that the random environment which was generated using a normal distribution persisted parallel to the motifs even after the digitisation process. This indicates that digitisation adheres to the lower bounding rule. But most importantly, this ability generates a Gaussian distribution by arranging the clusters indicate that the DNN has known that the random environmental conditions are derived from a normal distribution.

Even though the DNN was able to detect the distribution, it is not clear whether it was able to predict the random conditions. If the DNN could predict the random conditions, the clusters should indicate that the DNN response also behaved similarly to the environmental conditions. In the next section, we attempt to derive the distribution of DNN response by mapping it against the environmental conditions.

2) Predicting Random Data

Giving a specific treatment for each identified environment condition was prevailing throughout the decision rationale of the DNN. To visualise this, a distribution of DNN responses parallel to the above distribution of environmental conditions is constructed. The analysis confirms that the strategies applied by the DNN also follow a normal distribution. This implies that the DNN has been able to predict the random conditions (see Figure 13). The stable values of the DNN response indicate that it managed the conditions proactively. Hence, we can conclude that DNN effectively predicted environmental conditions.

VI. CONCLUSIONS AND FUTURE DEVELOPMENTS

Predicting random data is an excellent feature that DNN has to offer for autonomous operations in unpredictable en-

vironmental conditions. Explaining the process of the DNN increases the confidence of the autonomous transportation industry to replace fail-safe systems with more robust autonomous edge systems. This paper introduces a time series data mining mechanism to describe the decision rationale of a DNN that predicts random environmental conditions. The results concluded that the DNN acquires knowledge about the distribution of the random conditions and optimises stable responses that can withhold all possible random conditions.

In the process of discovering the secrets of the DNN, this work contributes to the research field of time series data analysis with a novel-zero-overlapping motif discovery process and a novel clusterable cross-comparable view generation technique. The cross-comparable view paved the way to describe the DNN process. This became possible due to the novel digitisation technique which is coherent and obeys the lower bounding rules. The digitisation technique can be used in other applications which require clustering concurrent data such as in financial market analysis and weather predictions. Subsequent research may further compare this novel method with other sampling techniques in other domains such as Monte Carlo methods and Gibbs Sampling [25].

Future research includes utilising the digitisation technique presented in this paper to provide a distributed cooperative learning system where the devices learn from each other in close collaboration with a central cloud. These potential developments will consider improving the learning mechanism by identifying various stages of the DNN and then distributing the learning functionality so that the whole autonomous process is distributed across system entities within the network. The devices thus become both consumers and producers of the information in addition to producing local intelligence that coordinates with a central cloud and incrementally influences the decisions of collaborators in the network. The proposed approach will lead to an AI-based dynamic and distributed control system that is trustworthy and explainable and can adapt to both local and global conditions in almost real-time.

REFERENCES

[1] Tuohy S., Glavin M., Hughes C., Jones E., Trivedi M. Kilmartin L. 2015. Intravehicle networks: A review. IEEE Transactions on Intelligent Transportation Systems 16(2), pp. 534–545, doi:10.1109/TITS.2014.2320605

[2] Zall Kusek, J., Görgens Prestidge, M., Hamilton, B. C. (2013). Fail-safe management: Five rules to avoid project failure. The World Bank.

[3] Nunus, A. 2019. Driverless cars: researchers have made a wrong turn. Nature ISSN 1476-4687 (online). doi: 10.1038/d41586-019-01473-3.

[4] Dassanayake, P.M., Anjum, A., Manning, W. and Bower, C., 2019, December. A Deep Reinforcement Learning based Homeostatic System for Unmanned Position Control. In Proceedings of the 6th IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (pp. 127-136).

[5] Gunning, D. (2019). DARPA's explainable artificial intelligence (XAI) program. Proceedings of the 24th International Conference on Intelligent User Interfaces.

[6] Fu, T.C., 2011. A review of time series data mining. Engineering Applications of Artificial Intelligence, 24(1), pp.164-181.

[7] Keogh, E. and Lin, J., 2005. Clustering of time-series subsequences is meaningless: implications for previous and future research. Knowledge and information systems, 8(2), pp.154-177.

[8] Lin, J., Keogh, E., Lonardi, S., Patel, P., 2002. Finding motifs in time series. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2nd Workshop on Temporal Data Mining, pp. 53–68.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2021.3083990, IEEE Transactions on Network Science and Engineering

IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING SPECIAL ISSUE ON ARTIFICIAL INTELLIGENCE / MACHINE LEARNING ENABLED RECONFIGURABLE WIREL

[9] Gao, Y. and Lin, J., 2019. Discovering Subdimensional Motifs of Different Lengths in Large-Scale Multivariate Time Series. arXiv preprint arXiv:1911.09218.

[10] Zhang, Y., Gallimore, M., Bingham, C., Chen, J., & Xu, Y. (2016). Hybrid hierarchical clustering—piecewise aggregate approximation, with applications. International Journal of Computational Intelligence and Applications, 15(04), 1650019.

[11] Coppens, Y., Efthymiadis, K., Lenaerts, T., Nowé, A., Miller, T., Weber, R. and Magazzeni, D., 2019. Distilling deep reinforcement learning policies in soft decision trees. In Proceedings of the IJCAI 2019 Workshop on Explainable Artificial Intelligence (pp. 1-6).

[12] Kim, J., & Canny, J. (2017). Interpretable learning for self-driving cars by visualizing causal attention. In Proceedings of the IEEE international conference on computer vision (pp. 2942-2950).

[13] Sow, D.; Turaga, D.; Schmidt, M. Mining of Sensor Data in Healthcare: A Survey. In Managing and Mining Sensor Data; Aggarwal, C.C., Ed.; Springer: Berlin, Germany, 2013; pp. 459–504

[14] Sun, Y., Li, J., Liu, J., Sun, B., & Chow, C. (2014). An improvement of symbolic aggregate approximation distance measure for time series. Neurocomputing, 138, 189-198.

[15] Li, H. (2020). Time works well: dynamic time warping based on time weighting for time series data mining. Information Sciences.

[16] Berlin E. and Van Laerhoven K., 2012 "Detecting leisure activities with dense motif discovery," in Proceedings of the 2012 ACM Conference on Ubiquitous Computing. ACM, pp. 250–259.

[17] McGovern, A., Rosendahl, D. H., Brown, R. A., & Droegemeier, K. K. (2011). Identifying predictive multi-dimensional time series motifs: an application to severe weather prediction. Data Mining and Knowledge Discovery, 22(1-2), 232-258.

[18] Wang L. and et al., 2010, "A tree-construction search approach for multivariate time series motifs discovery," Pattern Recognition Letters, vol. 31, no. 9, pp. 869–875.

[19] Gao, Y., & Lin, J. (2019). Discovering Subdimensional Motifs of Different Lengths in Large-Scale Multivariate Time Series. In 2019 IEEE International Conference on Data Mining (ICDM) (pp. 220-229). IEEE.

[20] Yeh C-C. M., Kavantzas N., and Keogh E., 2017. "Matrix profile vi: Meaningful multidimensional motif discovery," in Data Mining (ICDM), IEEE International Conference on. IEEE, pp. 565–574.

[21] Mehdi Khashei, Mehdi Bijari,(2011). A novel hybridization of artificial neural networks and ARIMA models for time series forecasting. Appl. Soft Comput. 11(2): 2664-2675.

[22] Keogh E., 2002 Exact Indexing of Dynamic Time Warping. In: Proc. of the 28th Very Large Databases Conf. (VLDB), Hong Kong, China, August 20-23, 2002, pp. 406-417

[23] Han, J., Pei, J. and Kamber, M., 2011. Data mining: concepts and techniques. Elsevier.

[24] Kumar, N., Lolla, V.N., Keogh, E., Lonardi, S., Ratanamahatana, C.A. and Wei, L., 2005, April. Time-series bitmaps: a practical visualization tool for working with large time-series databases. In Proceedings of the 2005 SIAM international conference on data mining (pp. 531-535). Society for Industrial and Applied Mathematics.

[25] Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Transactions on pattern analysis and machine intelligence, (6), 721-741.

[26] Ali, M., Anjum, A., Rana, O., Zamani, A.R., Balouek-Thomert, D. and Parashar, M., 2020. RES: Real-time video stream analytics using edge enhanced clouds. IEEE Transactions on Cloud Computing.

[27] Kiani, S.L., Anjum, A., Knappmeyer, M., Bessis, N. and Antonopoulos, N., 2013. Federated broker system for pervasive context provisioning. Journal of Systems and Software, 86(4), pp.1107-1123.

[28] Van Lingen, F., Thomas, M., Azim, T., Chitnis, I., Anjum, A., Bourilkov, D., Kulkarni, M., Steenberg, C., Cavanaugh, R.J., Bunn, J. and Ukin, J., 2005. Grid enabled analysis: architecture, prototype and status.

[29] Van Lingen, F., Thomas, M., Azim, T., Anjum, A., Bunn, J., Steenberg, C., Legrand, I. and Newman, H., 2005. The Clarens Grid-enabled Web Services Framework: Services and Implementation.

[30] Hasham, K., Peris, A.D., Anjum, A., Evans, D., Gowdy, S., Hernandez, J.M., Huedo, E., Hufnagel, D., van Lingen, F., McClatchey, R. and Metson, S., 2011. CMS workflow execution using intelligent job scheduling and data access strategies. IEEE Transactions on Nuclear Science, 58(3), pp.1221-1232.

Priyanthi M. Dassanayake Obtained her Bachelors degree from University of Kelaniya, Sri Lanka and her Masters Degree from University of Hertfordshire, UK. Mrs Dassanayake currently is a doctoral candidate at the University of Leicester under the supervision of Prof. Ashiq Anjum. Her research is centered on the Autonomous Dynamic Positioning in a Big data analytics project funded by Rolls Royce.

Professor Ashiq Anjum is a Professor of distributed at the University of Leicester, UK. Previously he was the Director of the Data Science Research Centre at the University of Derby, UK. His areas of research include data-intensive distributed systems and high-performance analytics platforms for continuous processing of streaming data.

Ali Kashif Bashir is with School of Computing and Mathematics, Manchester Metropolitan University, United Kingdom. He is also attached to School of Electrical Engineering and Computer Science, Islamabad (NUST) as an Adjunct Professor. He is a senior member of IEEE, USA and Distinguished Speaker of ACM, USA.

Joseph Bacon has lead numerous computer and data science-dependent start-ups and scale-ups across the globe. Joe has held various senior positions in Rolls-Royce including Chief of Engineering Systems and Global Innovation Manager. His current venture Make IT Different Ltd, is focused on combining Computer, Data, Systems Engineering and Human Sciences to deliver more effective, efficient and secure systems, whilst working closely with academia.

Rabia Saleem obtained an MSc degree in mathematics from University of Engineering and Technology, Lahore. Ms Rabia is a doctoral candidate at the University of Derby, UK and her research is centred around Explainable Artificial Intelligence (XAI) models,Deep neural networks (DNNs) and mathematical modelling of DNNs.

Professor Warren Manning gained his PhD from the University of Leeds in the field of AI and Control Systems. He continued his research activity at Leeds working on Intelligent Vehicle Motion control, especially in chassis control, advanced driver assistance systems and state/parameter estimation. He is currently Provost for Innovation and Research at the University of Derby.