Fuzzy Hot Spot Identification for Big Data: An Initial Approach

Rebecca Tickle[1], Isaac Triguero[1], Grazziela P. Figueredo[2], Ender Özcan[1], Mohammad Mesgarpour[3], Robert I. John[1]

[1]Automated Scheduling, Optimisation and Planning (ASAP) Research Group
[2]The Advanced Data Analysis Centre
School of Computer Science, University of Nottingham, UK
[3]Microlise, Farrington Way, Eastwood, Nottingham, UK
Email: {Rebecca.Tickle, Isaac.Triguero, Grazziela.Figueredo, Ender.Ozcan, Robert.John}@nottingham.ac.uk

***Abstract*—Hot spot identification problems are present across a wide range of areas, such as transportation, health care and energy. Hot spots are locations where a certain type of event occurs with high frequency. A recent big data approach is capable of identifying hot spots in a dynamic manner, through the processing of large volumes of sensor data arriving as a stream. However, the method may produce imprecise results due to its crisp interpretation of hot spot locations and reliance on a fixed hot spot radius value. This paper presents an initial approach to addressing this shortcoming through incorporating the concept of fuzzy hot spots into the process. Experimental results on large real-world transportation datasets demonstrate the improved way in which this approach handles uncertainty in the definition of hot spots, and highlight promising future research areas for further application of fuzzy systems to the hot spot identification problem.**

***Index Terms*—Hot Spot Identification, Fuzzy Logic, Big Data, Instance Selection, Data Streams**

I. Introduction

The process of hot spot identification (HSID) aims to discover hot spot locations, defined as areas with a high likelihood of occurrence of a given type of event. The HSID problem is applicable in a variety of contexts, such as transportation, health care and energy [1], [2]. In this work, we use transportation as a real-world case study, where the problem is to identify hot spots from heavy goods vehicle (HGV) data. A large portion of businesses and government sectors across the globe depend on HGVs for procurement and delivery of goods and services. Due to the importance of HGVs in the economy, there are great efforts to reduce their incident numbers. One of those efforts regards vehicle monitoring via telematics. HGVs are fitted with sensors that continuously generate large volumes of data on their status, location and incidents that occur within a journey. Detecting HGV incident hot spots allows transport companies and the government to trigger safety measures such as road repair, education, reward programs or law enforcement. In order to effectively exploit the available incident data and successfully identify hot spots, methods capable of handling very large volumes of data in a timely manner, with the adaptability to respond to changes in road or traffic conditions over time, are required.

Traditional approaches to HSID use statistical methods and historical incident data to determine hot spots [1]. However, these methods do not scale well to cases where large volumes of data are under consideration. More recently, data mining techniques such as clustering have been applied [3], but may result in the generation of invalid, elliptical hot spots. In recent work, we have proposed the use of instance selection techniques, usually employed to pre-process data prior to data mining tasks [4], for the task of HSID. In [5] and [6], an immune-inspired instance selection technique [7] was applied to find hot spots in large telematics datasets. Experimental results showed the success of this method (SeleSup-HSID) as a HSID approach. This approach was later extended [8], adapting the methodology in order to make it suitable for processing large, dynamic data streams.

While SeleSup-HSID and its later versions are able to provide timely and informative results in both the static and streaming big data scenarios, they do not address the inherent vagueness in the definition of a hot spot area or distance range. SeleSup-HSID requires a fixed hot spot radius to be provided, and a hot spot is therefore expected to indicate incidents within a pre-defined mileage range. This results in a single interpretation of hot spots being applied to all road locations despite not necessarily being suitable in all cases. In order to overcome this limitation, we consider the introduction of ideas from fuzzy systems into our instance selection-based HSID method. There are some fuzzy instance selection methods present in the literature; however, these are usually designed for use on datasets containing instances that are labelled with respect to the class they belong to [9], with some also focusing on selecting those instances situated close to class boundaries [10]. As HSID is an unsupervised data mining task, these existing methods are not applicable to our problem.

The aim of this paper is to provide an initial study of how HSID can be improved through the addition of fuzzy techniques. We first update our definition of hot spots, utilising the ideas of fuzzy sets. We subsequently use fuzzy inference to enable the identification of hot spots with varying radiuses, as well as to provide a more informative output. Experimental results on real-world transportation datasets show that the introduction of fuzziness is a promising area for further improvement of the HSID process. The remainder of this paper is organised as follows. In Section II we provide relevant background information and related work for the HSID problem, as well as briefly describing our previous non-fuzzy HSID method. In Section III we propose an initial approach to incorporating fuzzy concepts into the HSID process. In Section IV we present an experimental study analysing the behaviour of the proposed fuzzy method. Finally, in Section V we discuss the conclusions and identify opportunities for future work.

II. Background

In this section, we first describe the transportation HSID problem in detail, followed by a discussion of the related work for HSID and data mining using fuzzy techniques. Finally, we provide an overview of our previous immune-inspired HSID method, which is the basis of the initial fuzzy approach.

A. Problem Description

The HSID problem for transportation can be defined as follows: given a dataset containing vehicle incident locations, road areas where there is a high frequency of incident occurrence should be determined. Due to the large volumes of telematics data generated from HGV sensors, the solution must be capable of providing identified hot spots in a timely manner. The constraints for establishing hot spots are that incidents must occur on the same road, and have similar bearings.

The problem defined thus far is successfully addressed with SeleSup-HSID and its subsequent extensions. However, a limitation of the current solution is that it detects hot spots based on a fixed mileage range. Depending on the value for this parameter, a much smaller or larger number of indicators of high road incidents might occur, as illustrated in Figure 1. A higher mileage range may result in missed, potentially dangerous, areas, such as that highlighted by the red ellipse. Conversely, a smaller mileage range will produce redundancy in the identified hot spots, such as those inside the blue ellipse.

We now define the problem to include a requirement that the solution should account for the vagueness in the characterisation of a hot spot. Note that the definition only states that *areas of high likelihood of incident occurrence* should be identified, with no specification of the size of these areas. We find that by fixing the radius of hot spots, we risk providing either too much redundant information, or too little detail on the hot spot locations. It is therefore desirable to move towards a HSID method that does not create hot spots of a single, fixed mileage, but is instead capable of self-organising to the distribution of incidents present in the data and handles the uncertainty in the definition of hot spots.

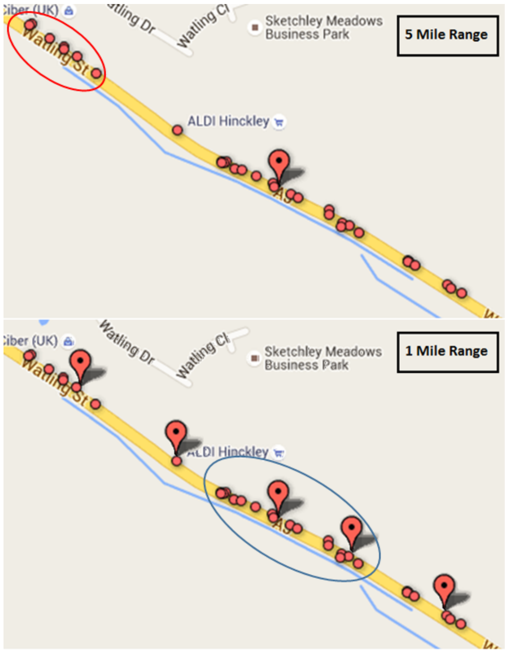

Fig. 1. Hot spots identified with fixed mileage ranges, demonstrating missed hot spots (mileage=5) and redundant hot spots (mileage=1).

B. Related Work

Previous approaches to HSID have employed statistical methods [1] and clustering techniques [3], [11]. The shortcomings of these methods lie in the fact that they may produce invalid results, require historical data or a predefined number of hot spots, and are not suitable for big data processing. Detailed reviews of the existing methods for HSID can be found in [5], [8]. The literature on employing concepts of fuzzy logic for HSID is limited. The neuro-fuzzy approach proposed in [12] uses an adaptive neuro-fuzzy inference system to process road features and environmental data to produce a hazard level for certain road locations. Their method requires road locations under investigation to be segmented, and does not generate fuzzy hot spots but rather assigns a hazard level to the individual road segments through fuzzy inference. In [13], fuzzy C-means clustering is first applied over pre-defined locations to obtain hot spot centres, after which each hot spot is assigned a safety level using fuzzy inference. Both the above approaches are not suitable for identifying hot spots over a wide geographical area, due to their requirements for specific data about the locations in question making them difficult to generalise.

C. Overview of Streaming SeleSup HSID

In this subsection we provide a high-level description of the streaming SeleSup-HSID algorithm, based on the earlier works on SeleSup-HSID in [5] and [6], and used here as the method into which we introduce fuzzy concepts. A full description can be found in [8] and our implementation is available on GitHub[1] (using Apache Spark Streaming [14]). The method is designed to work on a large data stream of incident data, processed as a sequential chain of microbatches containing incidents from a given time interval. The algorithm must be provided with a mileage range, defining the radius of hot spots. There are three stages, all of which are executed for every microbatch of incidents that arrive:

[1] https://github.com/beccatickle/PAS-HSID

- **Stage 1:** During the first stage, hot spots returned from the previous time interval are used to reduce the latest microbatch of incidents. If an incident is within mileage range of an existing hot spot, then it is considered to be reduced by that hot spot and can be deleted.
- **Stage 2:** Any incidents not reduced during Stage 1 potentially represent hot spots that have only recently appeared. The second stage aims to identify these new hot spot locations by calculating distances between these leftover incidents.
- **Stage 3:** The final stage updates the hot spot fitness values using the following equation:

$$FV_k^T = FV_k^{T-1} \cdot (1 - dr) + n_k^T \tag{1}$$

Where FV_k^{T-1} is the fitness value of hot spot k at the previous time interval, n_k^T is the number of incidents that hot spot k has reduced during the current time interval, and dr is the decay rate (used to control how fast hot spots will disappear after a period of time with no incidents occurring). Following this calculation, any hot spot for which FV_k^T is less than a specified deletion threshold is considered to have faded enough to no longer be a hot spot. Such hot spots are deleted. The resulting set of hot spots is then passed into Stage 1 of the next time interval, to be used to reduce the next microbatch of incidents.

The streaming SeleSup-HSID algorithm produces a set of hot spots that dynamically changes over time in response to the incidents occurring. However, it suffers from the issue of having a fixed mileage range, as previously described in Subsection II-A. In this paper, we develop a preliminary fuzzy HSID approach, in order to investigate whether the introduction of fuzzy concepts enables the implementation of a more general HSID method.

III. Fuzzy Hot Spot Identification

Our initial fuzzy HSID method is an extension of the streaming SeleSup-HSID algorithm, and follows the same three stages that are described in Section II-C. However, we update Stages 1 and 3 to introduce the concept of fuzzy hot spots (Subsection III-A), and use fuzzy inference to both vary the radius of hot spots and provide additional information about the nature of the identified hot spots (Subsection III-B).

A. *Defining Fuzzy Hot Spots*

In the original SeleSup-HSID algorithm, hot spots are viewed as disjoint sets $H_1 ... H_K$ where K is the number of hot spots that have been found. Incidents are reduced by either one or none of these hot spots; if an incident is reduced by a H_k it becomes a member of the associated set. Note that these sets are not actually maintained, as instances are discarded once they are reduced by a hot spot, but are rather used here to illustrate the concepts. We identify two problems with this crisp interpretation of hot spots:

- The radiuses of several hot spots may overlap, due to the appearance of new hot spot locations over time. When an incident i falls within a region of overlap, SeleSup-HSID will use only one of the hot spots to reduce i. Therefore, i will not contribute to any other hot spots, despite being located within their radius.
- For every incident that a hot spot H_k reduces, the fitness value FV_k, representing the strength of H_k, is incremented by one. Incidents located close to the edge of the mileage range contribute the same amount to the fitness value as those situated close to the hot spot centre.

We propose an alternative definition of hot spots as fuzzy sets, in order to alleviate these problems. In fuzzy C-means clustering [15], instances can belong to multiple clusters with varying degrees of membership. We use this idea of multiple membership when defining fuzzy hot spots. During the process of HSID, fuzzy hot spots $FH_1 ... FH_K$ are identified. When allocating incidents to the existing hot spots, they can be reduced by one, multiple or none of these fuzzy hot spots. The membership of an incident i in a fuzzy hot spot FH_k is given by the membership function $\mu_k(d_i)$ where d_i is the Haversine distance [16] between i and the centre of FH_k. The membership function for FH_k is defined as a Gaussian function centred on zero, chosen as its shape can be easily controlled by adjusting the standard deviation parameter:

$$\mu(d_i) = \begin{cases} \exp\left(\frac{-1}{2}\left(\frac{d_i}{\sigma_k}\right)^2\right) & d_i \leq maxMiles \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Where $maxMiles$ is the maximum radius of any hot spot (provided as a parameter) and σ_k is the standard deviation, calculated based on the density of FH_k (described in Subsection III-B). Note that d_i will always be positive.

We say that i is reduced by FH_k if $\mu_k(d_i)$ is greater than a given confidence threshold, $confTh$. If an incident is reduced by *at least one* fuzzy hot spot, then it is discarded at the end of the current time interval. We then define n_k, the value to increase the fitness value of FH_k by (see Equation 1), as the sum of the membership values of all incidents reduced by FH_k in this time interval:

$$n_k = \sum_{j=1}^{J_k} \mu_k(d_j) \tag{3}$$

Where J_k is the number of incidents reduced by FH_k.

In the original streaming SeleSup-HSID, n_k was simply defined as the number of incidents reduced by H_k. By defining n_k based on the membership values, incidents that we are more certain belong to the hot spot (that is, are closer to the hot spot centre) will contribute more to the fitness value than those where the membership is less certain. Allowing incidents to be reduced by multiple fuzzy hot spots ensures that in overlapping regions incidents will contribute to the fitness values of all hot spots they lie in range of.

By implementing fuzzy hot spots, we have changed how Stage 1 of the streaming HSID algorithm is performed. Algorithm 1 displays the pseudocode for the fuzzy Stage 1.

Algorithm 1: Stage 1 - ReduceWithFuzzyHotSpots

```
Require: HotSpots; Incidents; $ConfTh$
  forall HotSpots do $n_k = 0$;
  for all $i$ in $Incidents$ do
    for all $k$ in $HotSpots$ do
      $d_i \leftarrow$ calculate distance between $k$ and $i$
      $m \leftarrow$ calculate $\mu_k(d_i)$
      if $m \geq ConfTh$ then
        $Incidents \leftarrow Incidents - i$
        $n_k$ += $m$
      end if
    end for
  end for
```

B. Fuzzy Inference for Controlling Hot Spots

In the previous subsection we defined fuzzy hot spots and how the membership of incidents within a fuzzy hot spot is calculated. In this subsection, we consider how we can remove the need for a fixed mileage range, through dynamically controlling the membership functions of individual hot spots. As an initial approach, we propose to adjust the membership function $\mu_k(d)$ of a fuzzy hot spot FH_k based on the density of FH_k. The density of a fuzzy hot spot is here used to describe the distribution of the incidents that contribute to that hot spot. Intuitively, if the hot spot consists of a large number of incidents that are all located close to the hot spot centre, then it is a dense hot spot. In this case, the radius should be restricted in order to give a more precise location for the hot spot. Conversely, if the incidents tend to be situated further from the centre, then it is a sparse hot spot. The radius of sparse hot spots should be wider than that for dense hot spots, as the road area that the incidents span is larger.

As described in the previous subsection, the fuzzy hot spot membership functions are Gaussian and the closer an incident is to a hot spot centre, the greater its membership value. We control the shape of the function by changing the standard deviation, depending on the density of the hot spot. For denser hot spots, the standard deviation should be reduced; consequently, incidents will need to be closer to the hot spot centre in order to obtain a membership value greater than the confidence threshold and contribute to the hot spot. For sparser hot spots, the standard deviation should be increased. Figure 2 illustrates the differences between example membership functions of dense and sparse hot spots.

The overall effect is an automatic expansion or shrinking of the effective radius of individual hot spots, producing a hot spot identification method that self-organises, in terms of both hot spot locations and their mileage ranges, depending on the distribution of incidents. Furthermore, by recalculating the density and standard deviation of fuzzy hot spots at each time interval of the stream, the radius of hot spots is adjusted over time in response to newly arriving incidents.

In order to implement these dynamic membership functions, we need to define how to determine the density of hot spots, as well as how to use the density to calculate the standard deviations. For this initial work on fuzzy hot spot identification, we use a simple zero-order TSK fuzzy inference system [17] in order to establish the density of the hot spots.

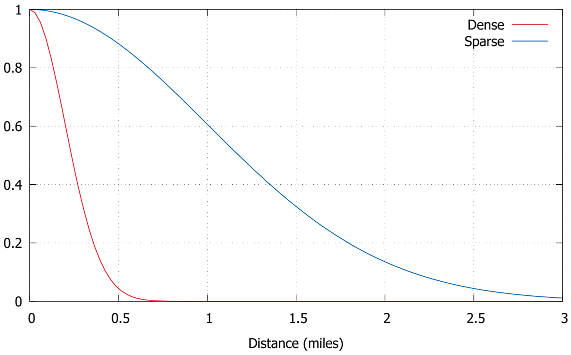

Fig. 2. Example membership functions of dense and sparse hot spots.

We establish a small rule base, with input variables $numIncidents$ (number of incidents added to a hot spot in the current time interval) and $avgDistance$ (average distance of the added incidents from the hot spot centre). The output variable is the density of the incidents recently added to the hot spot. These rules were developed through analysis of the distribution of incidents within hot spots identified by the original SeleSup-HSID algorithm. Table I summarises the rule base, with all rules taking the form "IF $numIncidents$ is x AND $avgDistance$ is y THEN $density$ is z".

TABLE I
Summary of the fuzzy rule base. The rule consequents represent the density of the hot spot.

		numIncidents		
		low	medium	high
avgDistance	low	medium	high	very high
	medium	very low	low	low
	high	very low	very low	low

For a given hot spot FH_k, the output of combining these rules is a single value in the range $[0, 1]$. This value is considered the density proportion, p_k^{new}, of incidents added to FH_k in the current time interval. However, this algorithm is designed for use with dynamic data streams, and so we also need to take into consideration the previous nature of FH_k prior to calculating an updated standard deviation. Therefore, in order to obtain the density of FH_k at the current time interval T, we calculate the following weighted average:

$$p_k^T = \frac{(0.5 \cdot p_k^{T-1}) + p_k^{new}}{1.5} \quad (4)$$

Where p_k^{T-1} is the density of FH_k at the previous time interval. Thus, the impact of old incidents is incorporated into the current density, with their influence gradually fading over time. We directly use p_k^T to calculate a new standard deviation of FH_k as follows:

$$\sigma_k^T = \sigma_{min} + ((1 - p_k^T) \cdot (\sigma_{max} - \sigma_{min})) \quad (5)$$

Where σ_{min} and σ_{max} are lower and upper bounds of the standard deviation. For the purposes of this study, we set

$\sigma_{min} = 0.15$ and $\sigma_{max} = 2.75$. These values were calculated in order to produce a reasonable range of hot spot radiuses, roughly between 0.1 and 2 miles.

We also utilise p_k^T to provide additional information on the nature of identified hot spots to stakeholders in an easily understandable manner. We label hot spots as either $Dense$ or $Sparse$, using the following rule:

$$label_k^T = \begin{cases} Dense & p_k^T \geq 0.5 \\ Sparse & \text{otherwise} \end{cases} \quad (6)$$

The calculation of an updated density and σ_k for each hot spot occurs during Stage 3 of the algorithm, in order to define the membership functions for the next time interval.

IV. Experimental Study

Two sets of telematics data for HGV incidents over a three-month period within the UK are investigated. We split these datasets in two different ways: into ten equally sized batches, and into day-long batches. Table II shows the average number of incidents per batch for both datasets.

TABLE II
Average number of incidents per batch for the used datasets

Dataset	Day	Equal	Total
Harsh Braking (HB)	2298	21369	213696
Contextual Speeding (CS)	7762	72187	721878

We compare the fuzzy approach with the non-fuzzy streaming SeleSup-HSID algorithm, both implemented under Apache Spark Streaming [14], a big data processing framework. For the non-fuzzy version, we use *dr*=0.3, *delTh*=1.9, *hsTh*=3 and mileage ranges of 0.5 and 5.0 for harsh braking and contextual speeding data respectively, as was done in [8]. For the fuzzy method, we use *maxMiles*=2, *dr*=0.3, *delTh*=1.2 and *hsTh=3*. For the non-fuzzy approach, *delTh* is chosen so that newly-initialised hot spots must encompass at least two incidents to avoid deletion. In order to achieve this same effect in the fuzzy version, where fitness values are calculated differently, we choose a lower value for *delTh*. We also analyse the impact of using different confidence thresholds on the behaviour of the fuzzy approach, specifically with the values $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. The experiments have been carried out in a single node with an Intel(R) Xeon(R) CPU E5-1650 v4 processor (12 cores) at 3.60GHz, and 64 GB of RAM. We have used the Cloudera open-source Apache Hadoop distribution (Hadoop 2.6.0-cdh5.14.2) and Spark 2.0.0. In our experiments, we use 8 partitions.

Table III displays the results in terms of number of hot spots and runtime for both the original streaming SeleSup-HSID algorithm and the fuzzy version. The fuzzy version returns a much larger number of hot spots for the same hot spot threshold than the non-fuzzy version. This may be due to a general increase in fitness values, caused by incidents contributing to multiple hot spots rather than a single hot spot. An increase in the value of the confidence threshold tends to lead to an increase in the number of identified hot spots, as incidents require a higher degree of membership to be reduced by existing hot spots, and are therefore more likely to form new hot spots instead. A similar effect was observed in [6] following an increase in the defined mileage range. The difference in the fuzzy version is that within the set of identified hot spots, a variety of different radiuses will be present, depending on the distribution of incidents. The value of *ConfTh* therefore provides some control over the granularity of hot spots, whilst not being restricted to a fixed radius.

In terms of runtime, the comparison is made with *ConfTh*=0.1, which has the fastest execution time of the confidence thresholds due to the reduced number of hot spots it identifies. The fuzzy HSID approach is slower than the original SeleSup-HSID, which can be attributed to the increased number of comparisons performed between incidents and hot spots in Stage 1 of the algorithm. In SeleSup-HSID, because incidents are only reduced by up to a single hot spot, the algorithm stops searching as soon as a suitable hot spot is found. However, with the introduction of fuzzy hot spots, we need to calculate the membership of every incident within every hot spot to ensure that incidents contribute to each hot spot that they fall in range of. Despite this, the runtimes of the fuzzy version are still reasonable for the size of the datasets considered.

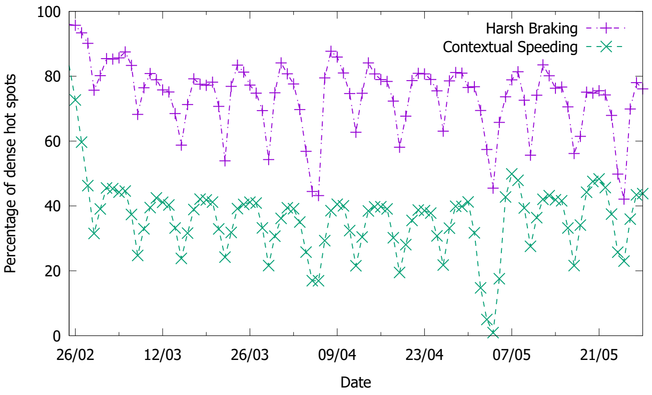

Fig. 3. Comparison of percentage of dense hot spots for Harsh Braking and Contextual Speeding data.

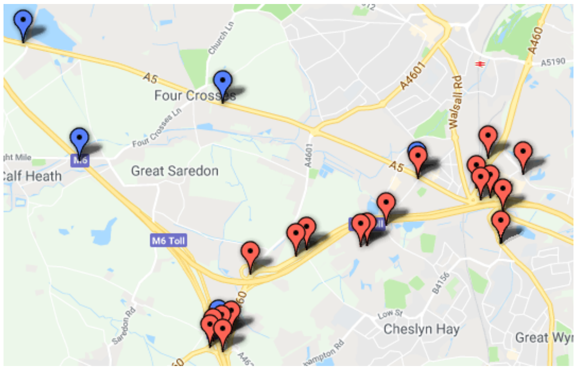

Fig. 4. Map showing dense (red markers) and sparse (blue markers) hot spots identified by our fuzzy HSID algorithm. Dense hot spots tend occur at road junction, while sparse hot spots occur on straighter stretches of road.

TABLE III
RESULTS FOR THE FUZZY AND NON-FUZZY VERSIONS OF STREAMING SELESUP-HSID, AVERAGED PER INTERVAL OF THE STREAM. NOTE THAT THE FUZZY APPROACH IS DENOTED AS FHSID-*ConfTh*, AND THE NON-FUZZY VERSION IS DENOTED AS HSID.

Dataset	**FV $\geq$ hsTh**						**Runtime (seconds)**	
	FHSID-0.1	FHSID-0.2	FHSID-0.3	FHSID-0.4	FHSID-0.5	HSID	FHSID-0.1	HSID
HB-day	334	336	339	341	340	261	1.676	0.975
HB-equal	4543	4620	4695	4775	4848	4358	9.114	6.353
CS-day	2163	2213	2277	2357	2448	1614	2.819	1.295
CS-equal	9377	9675	10037	10466	11158	5930	30.372	6.338

Figure 3 displays the difference in the number of dense hot spots found within the harsh braking and contextual speeding datasets, represented as a percentage of the total number of hot spots. The nature of harsh braking incidents is that they frequently occur in very specific locations, for example close to road junctions. Contextual speeding incidents will usually occur at multiple points along a long stretch of a road. We can see that the fuzzy HSID approach identifies a higher proportion of dense hot spots for the harsh braking dataset than contextual speeding, suggesting that the method is successfully adapting the radius of individual hot spots in response to the incidents that they contain. Further evidence of this is provided in Figure 4, which shows a sample of the hot spots identified within the harsh braking dataset. This demonstrates the improved self-organising nature of the fuzzy approach. Around road junctions, a large number of dense hot spots are identified, giving very precise locations for hot spots in these areas. Conversely, on straighter stretches of road where less precision is required, sparse hot spots are identified that encompass incidents over a larger distance. In Figure 3, the regular drops in the percentage of dense hot spots coincide with data from weekends. As discussed in [8], at these times there are fewer HGVs active, and therefore a reduction in the number of incidents occurring. Over the weekends, dense hot spots decay into sparse hot spots, resulting in the patterns seen in this figure.

Overall, these results demonstrate that this initial fuzzy approach, despite utilising simple techniques, is able to provide more precise and informative hot spots, with a minor increase in the processing time.

V. CONCLUSIONS

In this paper, we have presented a first approach for incorporating the ideas of fuzzy sets and fuzzy inference into an existing HSID method for big data, in order to improve the way in which it handles the inherent uncertainty of the problem. The experimental study shows that the use of simple fuzzy techniques has enhanced the self-organisation of the hot spots, removing the requirement of a fixed mileage range and enabling a more informative output to be presented to stakeholders. These results provide a foundation for future research into additional ways in which fuzzy logic can be utilised to continue improving HSID methods. For example, the fuzzy inference could be extended to include the dynamic aspects of the streaming algorithm, removing the need for the associated thresholds and further generalising the method.

ACKNOWLEDGMENTS

We would like to thank Dr Mohammad Mesgarpour and Matt Hague from Microlise for the support and for providing the large data sets that made this research possible.

REFERENCES

[1] W. Cheng and S. P. Washington, "Experimental evaluation of hotspot identification methods," *Accident Analysis & Prevention*, vol. 37, no. 5, pp. 870–881, 2005.

[2] B. Elen, J. Peters, M. van Poppel, N. Bleux, J. Theunis, M. Reggente, and A. Standaert, "The aeroflex: A bicycle for mobile air quality measurements," *Sensors (Switzerland)*, vol. 13, no. 1, pp. 221–240, 2013.

[3] T. K. Anderson, "Kernel density estimation and k-means clustering to profile road accident hotspots," *Accident Analysis & Prevention*, vol. 41, no. 3, pp. 359 – 364, 2009.

[4] S. García, J. Luengo, and F. Herrera, *Data Preprocessing in Data Mining*. Springer Publishing Company, Incorporated, 2014.

[5] G. P. Figueredo, I. Triguero, M. Mesgarpour, A. M. Guerra, J. M. Garibaldi, and R. I. John, "An immune-inspired technique to identify heavy goods vehicles incident hot spots," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 1, no. 4, pp. 248–258, 2017.

[6] I. Triguero, G. P. Figueredo, M. Mesgarpour, J. M. Garibaldi, and R. I. John, "Vehicle incident hot spots identification: An approach for big data," in *2017 IEEE Trustcom/BigDataSE/ICESS*, 2017, pp. 901–908.

[7] G. P. Figueredo, N. F. F. Ebecken, D. A. Augusto, and H. J. C. Barbosa, "An immune-inspired instance selection mechanism for supervised classification," *Memetic Computing*, vol. 4, pp. 135–147, 2012.

[8] R. Tickle, I. Triguero, G. P. Figueredo, M. Mesgarpour, and R. I. John, "PAS3-HSID: A dynamic bio-inspired approach for real-time hot spot identification in data streams," *Cognitive Computation*, 2019, in press.

[9] R. Jensen and C. Cornelis, "Fuzzy-rough instance selection," in *International Conference on Fuzzy Systems*, 2010, pp. 1–7.

[10] A. S. Alvar and M. S. Abadeh, "Efficient instance selection algorithm for classification based on fuzzy frequent patterns," in *IEEE 17th International Symposium on Computational Intelligence and Informatics (CINTI)*, 2016, pp. 319–324.

[11] J. Han, M. Kamber, and A. K. H. Tung, "Spatial clustering methods in data mining: A survey," in *Geographic Data Mining and Knowledge Discovery, Research Monographs in GIS*, H. J. Miller and J. Han, Eds. Taylor and Francis, 2001.

[12] M. Effati, M. A. Rajabi, F. Samadzadegan, and S. Shabani, "A geospatial based neuro-fuzzy modeling for regional transportation corridors hazardous zones identification," *International Journal of Civil Engineering*, vol. 12, 2014.

[13] Y. S. Murat and Z. Cakici, "An integration of different computing approaches in traffic safety analysis," *Transportation Research Procedia*, vol. 22, pp. 265 – 274, 2017.

[14] M. Zaharia, T. Das, H. Li, S. Shenker, and I. Stoica, "Discretized streams: An efficient and fault-tolerant model for stream processing on large clusters," in *Proceedings of the 4th USENIX conference on Hot Topics in Cloud Computing*, 2012, pp. 10–10.

[15] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, 1981.

[16] G. Van Brummelen, *Heavenly mathematics: The forgotten art of spherical trigonometry*. Princeton University Press, 2012.

[17] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.