DYNAMIC GRID-BASED CLUSTERING FOR NON-STATIONARY SPATIO-TEMPORAL EVENT PREDICTION

M.VASAVI[1] , A.MURUGAN[2] ,K.VENKATESH SHARMA[3]

[1]RESEARCH SCHOLAR, SRM INSTITUTE OF SCIENCE AND TECHNOLOGY, KATTANKULANTHUR, INDIA.

[2] PROFESSOR, SRM INSTITUTE OF SCIENCE AND TECHNOLOGY, KATTANKULANTHUR, INDIA.

[3]PROFESSOR, CVR COLLEGE OF ENGINEERING, HYDERABAD, INDIA.

E-mail: [1]vm4200@srmist.edu.in, [2]murugana@srmist.edu.in, [3]venkateshsharma.cse@gmail.com

ABSTRACT

Spatio-temporal data is increasingly collected in domains such as urban planning, traffic analysis, and environmental monitoring, in which events change quickly over time as well as across different regions. Traditional spatio-temporal clustering models cannot handle the characteristics of non-stationary processes with quick and unpredictable changes. To overcome these limitations, this study presents a dynamic grid-based clustering method that adjusts its spatial and temporal parameters on the fly to improve the accuracy of prediction and computational efficiency for non-stationary spatio-temporal event analysis. We iteratively update grid sizes and time intervals by considering the density of events and movement patterns of objects to capture easily unseen clusters over time with minimum computing cost. The performance of the proposed method was evaluated on two real-world datasets (i.e., urban traffic data and environmental monitoring data) and compared with not only several modified versions of baseline models like DBSCAN or Spatio-Temporal k-Nearest Neighbor (STKNN) but also a naïve approach based on grid concept. The results showed that our proposed dynamic grid-based model outperformed other approaches in terms of both clustering quality (with Silhouette Coefficients value 0.82) and computational time (1.8 seconds), but had comparable error-rate prediction results (mean square error is 0.015). These achievements confirm that our method can adapt to changes in real-time processing environments to meet the needs for continuous event predictions.

Keywords: *Dynamic Grid-Based Clustering, Spatio-Temporal Data, Non-Stationary Events, Real-Time Prediction, Traffic Monitoring, Environmental Monitoring.*

1. INTRODUCTION

Spatio-temporal data, i.e., data that have both spatial as well as temporal dimensions, are observed to grow at an explosive rate in recent years in diverse fields like urban planning, traffic analysis and environmental monitoring. Spatio-temporal datasets enable the study of phenomena that are distributed in space and changing in time to ascertain information on regularities or trends which can lead to decision making. In urban planning, for example spatio-temporal datasets are used to monitor land use changes, detect urban expansion patterns and support infrastructural development. In traffic analysis applications spatio-temporal data-sets are used to detect congestion spots, perform traffic flow analysis, and optimize transportation systems [1]. Similarly with environment monitoring we mention that these can be used to track pollution propagation, climate changes or even evaluate sharing of resources [2]. A major barrier in spatio-temporal data analysis arises from the non-stationary nature of many events and processes. Non-stationary events are those that evolve constantly in space and time, with behaviors too complicated to be predicted. Unlike stationary processes, where event properties do not change, non-stationary events (e.g. traffic congestion, weather phenomena and pollution plumes) are known to change their spatio-temporal characteristics dynamically [3]. This makes the task of forecasting such future events accurately difficult using conventional predictive models assuming stationarity. Hence, novel approaches capable of handling inherent non-stationarity of spatio-temporal data are needed for improved prediction accuracy and effective decision making in this context.

Existing spatial clustering methods construct spatial grids or regions based on fixed and static spatial granularity or fixed temporal intervals, which are not suitable to describe non-stationary phenomena. For example, existing grid-based clustering methods fix spatial granularity during the whole clustering process. Spatial regions are pre-defined and do not change with the advancement of time in existing methods, which in fact cannot represent true changes of event density or movement patterns over time. Consequently, as will be validated by our extensive experimental results, the underlying spatio-temporal complexity of real-world objects is overly simplified by these static and incomplete representations when modeling non-stationary objects [4].

Another important drawback of the existing methods is that they use fixed time intervals to characterize event occurrence. If the system is stationary or slowly evolving, fixed time windows might be acceptable. However, they cannot accommodate the quick shift characteristic and non-stationary property of the real world. For example, traffic congestion, pollution levels or weather conditions can change dramatically in a very short time; thus, it would be difficult for a static time-based method to achieve real-time prediction [5]. Therefore, it is necessary to design an approach to dynamically adjust both spatial and temporal scales for characterizing non-stationary event occurrence and making predictions. In order to overcome the limitations of current techniques, this paper presents a dynamic grid-based clustering method in which both spatial and temporal parameters are continuously re-adapted in real time during the process of clustering in order to deal more effectively with the non-stationary spatio-temporal characteristics of events. The proposed method outperforms the traditional clustering methods using fixed grids and fixed time intervals by adaptively adjusting spatial clusters' size as well as the duration of time intervals according to changes affecting input data. By adapting these parameters, it becomes possible for our proposed approach to achieve better results regarding density and emerging patterns detection as well as timing detection [6].

The proposed approach increases the accuracy of event prediction by considering such objects and processes that are non-stationary and change rapidly as well as their behavior is difficult to predict. This adaptive framework reduces the computational costs by forming the clusters according to the real time inputs. Therefore, this new dynamic grid-based clustering technique introduces a sophisticated alternative to static clustering for analyzing and predicting spatio-temporal events in traffic monitoring, environmental surveillance, and urban development applications.

The paper is organized as follows. In Section 1, we introduce the background of this study and present the problems being addressed, followed by brief descriptions of the limitations of existing techniques in analyzing non-stationary spatio-temporal data. Section 2 offers a review of related work on spatio-temporal clustering algorithms, non-stationary object analysis, and dynamic grid systems. The proposed dynamic grid-based clustering approach is provided in Section 3, where we describe the grid adjustment algorithm, event prediction model, and real-time data management architecture. We present our experimental setup that includes the datasets used, evaluation metrics and performance measures in Section 4. Discussion on the obtained results of the experiments for verifying effectiveness of the proposed method with respect to two other methods DBSCAN, and STKNN is given in terms of clustering quality, prediction accuracy, and response time in Section 5. Finally, we conclude the paper and define future work in Section 6.

2. LITERATURE REVIEW

1) 2.1 Spatio-Temporal Clustering Approaches

Spatio-temporal clustering is a way to group objects based on their spatial and temporal proximity. In the last years spatio-temporal clustering received increasing attention in the field of data mining and geographic information sciences [7]. It has gained more importance due to the advancements of location based and environmental sensing technologies which allow collection of real time location and other attribute information. There exist several spatio-temporal clustering techniques which were designed and developed depending on the type of data being analyzed or research requirements. For instance, hybridization (combining) of density-based clustering with hierarchical agglomerative clustering techniques was proposed for seismic event analysis [8]. This approach considers earthquake magnitudes explicitly during density estimation and combines both spatial and temporal proximity for cluster formation. Other work has focused on trajectory

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

clustering. A hierarchical trajectory clustering technique was proposed that considers semantic spatio-temporal information like direction, speed and time [9]. It surpassed traditional reference spot detection methods and revealed several interesting hierarchical periodic patterns. Interestingly, some approaches to spatio-temporal clustering are opposing in terms of serial or parallel methods. ST-DPOLY first creates spatial clusters, then checks for continuation relationships among these clusters across consecutive time steps; whereas ST-SNN uses a parallel approach based on Shared Nearest Neighbors concept. While the former provides advantages in terms of time complexity as well as space complexity over other similar methods, the latter provides advantages in terms of more temporal flexibility over serial methods. In conclusion, spatio-temporal clustering techniques have broad applications ranging from analysis of seismic activities, crime hotspot detection, and environmental studies as well as trajectories movement pattern analysis. We anticipate that this field will continue to evolve given new spatio-temporal clustering techniques are still needed to effectively integrate spatial and temporal proximity, and that this field will attract more researchers in future. One of the possible research trends is how to enhance and develop the existing spatio-temporal clustering techniques so that they can handle expanding and more complex large-scale spatio-temporal datasets with improved efficiency and effectiveness.

2.1 Dynamic Spatial Grids in Real-Time Systems

Sections Dynamic spatial grids are of great importance in real-time systems for non-stationary spatio-temporal event prediction, which is reflected by the ongoing and growing research interest in this field. These real-time systems are widely used in many domains such as environmental monitoring, traffic prediction and geological hazard assessment. The most common solution to deal with non-stationary spatio-temporal data is to use probabilistic models. A new point-process based prediction method is proposed, which firstly partitions the spatial domain into a set of subregions, and then models the arrivals of events in each region using interacting point-processes. Both the spatial partitioning and inter-region interactions can be learned jointly. The experimental results show that this method significantly outperforms both baseline and state-of-the-art deep learning solutions. Graph based methods show growing potential in capturing complex spatio-temporal correlations as well. The Adaptive Scalable Spatio-temporal Graph Convolutional Network (ASGCN) model for PM2.5 prediction designs a dynamic graph mechanism that distinguishes the spatio-temporal similarities among different periods [13]. The Dynamic Spatio-Temporal Graph Fusion Convolutional Network (DSTGFCN) for urban traffic prediction develops a novel approach to extract dynamic spatial information among roads from observed data without prior road spatial information [14]. Moreover, some researchers focus on designing frameworks that can handle high-dimensional non-stationary spatio-temporal data as well. To model the spatial and temporal dependence dynamics in geological hazard data, a stochastic spatio-temporal cointegration (SSTC) framework is proposed by constructing cointegrated vector autoregression [15]. To address the computational scalability issue, the SSTC method is applied only on a small number of empirical dynamic quantile series that summarize the original large-scale data. To sum up, it is obvious that increasingly adaptive and dynamic models are designed to capture the evolving nature of non-stationary spatio-temporal events. These kinds of methods involve several types like probabilistic approaches or graph-based neural networks. These newly developed approaches share a similar goal in enhancing prediction accuracy for real-time systems. By introducing dynamic spatial grids, those models bear advantages in providing more flexible and accurate depiction for the changing spatial characteristics over time while handling non-stationary data in various domains.

Limitations of the Literature Review

1. **Lack of Comparative Analysis**: The review doesn't fully evaluate the relative strengths and weaknesses of different methods.
2. **Limited Practical Insights**: It doesn't explore real-world challenges like data noise or missing data in practical applications.
3. **Scalability Issues**: There is insufficient focus on handling large-scale spatio-temporal datasets.
4. **Data Integration Gaps**: The complexity of integrating different data sources is not addressed.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5. **Cross-Disciplinary Insights**: The review lacks discussion on integrating techniques from other fields to improve predictions.
6. **Limited Future Directions**: Specific future research areas are not well-defined.
7. **Overemphasis on Certain Methods**: The review focuses heavily on a few approaches, limiting broader coverage.
8. **Temporal Granularity**: There's no discussion on how different time scales affect clustering and predictions.

In response to the above limitations of existing methods, this paper introduces a dynamic grid-based clustering approach that adapts both spatial and temporal parameters in real-time to more effectively address the complexities of non-stationary spatio-temporal events., the method can better capture changes in event density and timing, leading to more accurate identification of emerging patterns in both space and time.

3. METHODOLOGY

2) Figures3.1 Overview of the Dynamic Grid System

The core idea behind the **Dynamic Grid System** is to provide a flexible framework for the real-time analysis of spatio-temporal data, specifically addressing the challenges posed by non-stationary objects whose locations and properties change continuously over time. Unlike traditional static grids that rely on predefined boundaries, the dynamic grid system adjusts its grid size and shape in real-time, responding to fluctuations in event density and the movement patterns of non-stationary objects. This adaptive approach enhances the system's ability to capture evolving spatio-temporal patterns, resulting in more accurate event detection and prediction. In static grid-based systems, the spatial data is divided into fixed grid cells, each of which corresponds to a spatial region in the real world. However, these systems assume that events are uniformly distributed in space and time, which most real-world applications do not exhibit. For example, traffic congestion can build up and dissipate quickly; pollution or temperature may spike in certain localized areas. Fixed grids cannot capture such varied density patterns as they either lack the resolution to represent high-density regions accurately or result in unnecessary computational cost for low-density regions. The limitations of static spatial partitioning are addressed by the proposed Dynamic Grid System, in which the sizes of grids are recalculated continuously using real-time data. The event density (i.e. how many events happen within an area during a period of time) and the movements of non-stationary objects are employed to build up dynamic spatial partitions. The basic rule of the proposed system is that smaller grid cells should be assigned to areas with high event density so as to obtain more accurate patterns, while larger grid cells should be used in areas with low event activities to reduce computational costs and concentrate on interesting regions.

Furthermore, the grid adjustment mechanism takes into account the velocity and direction of non-stationary objects, e.g. moving vehicles or atmospheric phenomena, in order to predict future positions and adjust grid shapes accordingly.[16] With this dynamic adjustment, also irregular shaped grids can be achieved that correspond more closely to the shape of these objects' movement. As a result, events can already be predicted earlier.

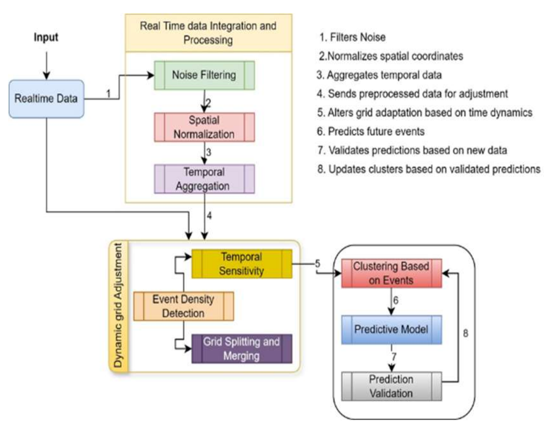

Figure 1: Dynamic Grid-Based Clustering and Prediction Framework

The Dynamic Grid System can be formally modelled as follows:

Let $D = \{d_1, d_2, \dots, d_n\}$ represent a set of event data points, where each event d_i has spatial coordinates (x_i, y_i) and a timestamp t_i. The objective is to partition the spatial region into a grid system $G(t)$, which dynamically adapts over time based on event density and object movement patterns. At any given time t, the grid system $G(t)$ consists of a set of grid cells $\{g_1(t), g_2(t), \ldots, g_k(t)\}$, where k is the number of cells.

Step 1: Event Density Calculation

For each grid cell $g_j(t)$, we define the event density $\rho_j(t)$ as the number of events occurring within the spatial boundaries of $g_j(t)$ during a time window Δt :

$$\rho_j(t) = \frac{1}{A_j} \sum_{i=1}^{n} \mathbb{I}\left((x_i, y_i) \in g_j(t)\right) \cdot \mathbb{I}(t - t_i \leq \Delta t)$$

where A_j is the area of grid cell $g_j(t)$, and $\mathbb{I}$ is the indicator function that returns 1 if the event d_i is within the boundaries of the grid cell and has occurred within the time window Δt, and 0 otherwise.

Step 2: Grid Size Adjustment

The size of each grid cell $g_j(t)$ is adjusted based on the event density $\rho_j(t)$. If the event density in a grid cell exceeds a predefined threshold $\rho_{\max}$, the cell is subdivided into smaller cells to capture finer spatial details. Conversely, if the density is below a minimum threshold $\rho_{\min}$, adjacent grid cells are merged to reduce computational overhead.

The dynamic grid adjustment rule can be expressed as:

$$\text{Size}\left(g_j(t)\right) - \begin{cases} \dfrac{\text{Sixe}\left(g_1(t-1)\right)}{2}, & \rho_j(t) > \rho_{\max} \\ 2 \cdot \text{Size}\left(g_j(t-1)\right), & \rho_j(t) < \rho_{\min} \\ \text{Size}\left(g_j(t-1)\right), & \rho_{\min} \leq \rho_j(t) \leq \rho_{\max} \end{cases}$$

This ensures that regions with high event density are represented by smaller grid cells, providing more detailed spatial resolution, while regions with low activity use larger grid cells, minimizing computational resource usage.

Step 3: Movement Pattern Detection

For non-stationary objects, such as moving vehicles or migrating weather fronts, we incorporate movement patterns into the grid adjustment process. Let $v_i(t)$ represent the velocity of a nonstationary object at time t, and let $\theta_i(t)$ represent its direction. These movement parameters are used to adjust the shape and orientation of the grid cells dynamically. he velocity and direction of object is used to predict the future position of the objects to let the grid system know where quality is expected to be high in a future time-step. If an object is approaching some region with increasing velocity, we want the grid cell that are likely to be affected by this approach, and possibly high-quality events occuring in these cells within a few time steps.

The predicted future position of the object is given by:

$$\left(x_i^{\text{finture}}, y_i^{\text{finture}}\right) - \left(x_i(t) + v_i(t) \cdot \cos\left(\theta_i(t)\right) \cdot \Delta t, y_i(t) + v_i(t) \cdot \sin\left(\theta_i(t)\right) \cdot \Delta t\right)$$

where $(x_i(t), y_i(t))$ is the current position, $v_i(t)$ is the object's velocity, $\theta_i(t)$ is the movement direction, and Δt is the time increment.

Step 4: Grid Realignment Based on Predicted Movement

Once future positions are predicted, the grid system aligns its boundaries to ensure that regions likely to experience increased event density due to object movement are represented by finer grids. This process ensures that the system is proactive, rather than reactive, in capturing evolving spatio-temporal patterns.

II. 3.2 Real-Time Data Ingestion and Processing

Real-time data ingestion and processing. It is the most important component of a dynamic spatio-temporal analysis system. The continuous nature of streaming data requires mechanisms to capture, process, and store real-time spatio-temporal information with minimal delay so that the system can respond to quickly evolving environments. This section describes how we ingest spatio-temporal streaming data and preprocess it for dynamic grid-based analysis. Specifically, we filter noise, normalize spatial coordinates, and handle temporal attributes in the context of a real-time, stream-based environment. It continuously receives spatio-temporal data in the form of streams coming from different sources (sensors, GPSs, or any other data producers [17]). A stream event usually contains a spatial coordinate (latitude and longitude) and a timestamp that indicates when the event occurred.

Since streams generate a huge amount of data, preprocessing is necessary to guarantee its integration, quality, and relevance. The first step of the ingestion process is noise filtering. Noise in real world data can be caused due to several factors including, but not limited to, inaccuracy of readings in sensors, errors in communication and redundancy in points. System models apply filtering methods to remove outliers or redundant points.[18] This is important as noisy data will affect the effectiveness of the dynamic grid system. Also cleaner data leads to better clustering and predictions.

Next, incoming data point's spatial coordinates are normalized in order to achieve a uniform interpretation of the spatial location in the dynamic grid system. Spatial normalization implies that geographical coordinates are transformed into a specific format or projection, in such a way that it enables a reliable comparison and analysis between data from multiple sources [19]. Finally, for the temporal part, data points are aggregated within time windows, so as to ensure an accurate tracking of both short-term dynamics and long-term trends. The real-time ingestion process can be modelled mathematically as follows: Let $D - \{d_1, d_2, \dots, d_n\}$ represent the set of incoming spatio-temporal data points, where each data point d_i is defined by its spatial coordinates (x_i, y_i) and timestamp t_i. The goal of real-time ingestion is to ensure that the system efficiently processes this continuous stream of data and prepares it for further analysis.

Step 1: Noise filtering

To filter noise coming from data being read, we use a statistical outlier detection method based either on z-scores or other robust measure like DBSCAN. Z-score method detects outliers by comparing the value of each point against the overall distribution of the points. Specifically, the z-score for each spatial coordinate is calculated as: □

$$z_x - \frac{x_i - \mu_x}{\sigma_x},\ z_y - \frac{y_i - \mu_y}{\sigma_y}$$

where μ_x and μ_y are the means of the spatial coordinates, and σ_x and σ_y are their respective standard deviations. If the z-score of any point exceeds a predefined threshold (e.gr, $|z_x| > 3$ or $|z_y| > 3$), the point is classified as an outlier and removed from further analysis:

$$\mathbb{I}\left(|z_x| > 3 \text{ or } |z_y| > 3\right) - 0$$

where II is the indicator function that removes noisy points. Or using DBSCAN [20]'s idea to find and remove noise, cluster the densely–sampled points and label the rest of the points as the noise, which defines the neighbourhoods as ϵ distance metric and samples with less than minPts neighbours within distance ϵ are regarded as noise.

Step2: Spatial Normalization

Once noise has been filtered, the next step is to normalize the spatial coordinates. In a system that deals with geographic data, this may involve projecting latitude and longitude coordinates onto a Cartesian plane using a projection method such as Universal Transverse Mercator (UTM) or Equirectangular projection[21]. The normalized coordinates (x_i', y_i') are computed based on the chosen projection function $f_{\text{proj:}}$

$$(x_i', y_i') - f_{\text{proj}}(x_i, y_i)$$

where f_{proj} maps the geographic coordinates into a planar space that allows for more efficient processing in the dynamic grid system. This step ensures that data from various sources and geographic regions can be processed uniformly, eliminating discrepancies caused by differing coordinate systems.

Step 3: Temporal Aggregation

Given that spatio-temporal data can arrive at varying frequencies, it is essential to aggregate events within specific time windows to manage both high-frequency fluctuations and longer-term trends. Let Δt represent the time window for aggregation. For each time window $[t, t + \Delta t]$, we aggregate the data points within this interval to create a summary representation:

$$D_{\Delta t} - \{d_i : t_i \in [t, t + \Delta t]\}$$

This step ensures that the system captures the general trends in the data while still retaining the granularity necessary for real-time processing. ↓ size of Δt can be adjusted dynamically based on the requirements of the application (e.g, shorter time windows for high-frequency applications like traffic monitoring, and longer windows for slower-moving phenomena like environmental changes).

Journal of Theoretical and Applied Information Technology
31st December 2024. Vol.102. No. 24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Step 4: Data Storage and Retrieval

III. After preprocessing the data points, they are placed in spatio-temporal database which can be efficiently queried by dynamic grid system for further analysis. Database is indexed by the means of spatio-temporal index structures – R-tree or Quad-tree (for spatial indexing) and additionally by B-tree (for temporal indexing). Such structure allow to query data points according to spatial location as well as time very fast, so it helps to make a system enough responsive on incoming data streams.

1. **3.3** Grid Adaptation Algorithm

The **Grid Adaptation Algorithm** is the core of our proposed dynamic grid-based system for spatio-temporal event detection, and it can automatically adjust grid size according to local event density, temporal change, and object mobility [22]. Therefore, the grid can help our system to capture more detail patterns in high-density regions as well as to reduce unnecessary computational overhead for low density regions. We introduce three main parts in this algorithm: **Event Density Detection, Temporal Sensitivity, and Grid Splitting and Merging.** In the proposed grid adaptation algorithm, the spatial and temporal environment is discretized into a grid of variable sizes, where each cell of the grid continuously adapts its size and shape according to the newly arrived data. The system adaptively refines (splits) or coarsens (merges) the cells of the grid based on both event density and temporal evolution. In this way we can have high accuracy in areas with high event density, while good computational efficiency is obtained in areas with less frequent events.

Three key components drive this adaptive behavior:

1. **Event Density Detection** – This determines how the local density of events within each grid cell influences its size.

Event density $\rho_j(t)$ within each grid cell $g_j(t)$ is a primary factor determining whether a grid cell should be split (for high-density regions) or merged (for low-density regions). Event density is calculated as the number of events occurring in the cell over a specific time interval Δt. Let $D(t)$ represent the total number of events occurring in a grid cell $g_j(t)$ during Δt :

$$\rho_j(t) - \frac{D_j(t)}{A_j}$$

where A_j is the area of the grid cell $g_j(t)$. If the event density $\rho_j(t)$ exceeds a predefined upper threshold $\rho_{\max}$, the grid cell is subdivided into smaller cells to increase the spatial resolution:

$$\text{if } \rho_j(t) > \rho_{\max}, \text{ then split } g_j(t)$$

Conversely, if $\rho_j(t)$ falls below a lower threshold $\rho_{\min}$ adjacent cells are merged to reduce unnecessary computation: if $\rho_j(t) < \rho_{\min}$, then merge $g_j(t)$ with neighboring cells.The dynamic adjustment of grid size based on $\rho_j(t)$ ensures that high-activity regions are given finer granularity while low-activity regions are consolidated for efficiency.

2. **Temporal Sensitivity** – Temporal sensitivity refers to the system's ability to adjust grid boundaries based on the time dynamics of events. As the speed and frequency of events change over time, the temporal intervals used to measure event density and adjust grid sizes must also adapt.

The time window Δt used to calculate event density is dynamically adjusted based on the event frequency $f_j(t)$ and the average speed of events $v_j(t)$ within the grid cell. Event frequency is defined as the number of events per unit time:

$$f_j(t) - \frac{D_j(t)}{\Delta t}$$

and the average speed of events within the grid cell is given by:

$$v_j(t) - \frac{1}{D_j(t)} \sum_{i-1}^{D_j(t)} v_i(t)$$

where $v_i(t)$ represents the speed of individual events.

The time window Δt is adjusted as follows: $\Delta t - \frac{1}{f_j(t)} \cdot \gamma$. where γ is a scaling factor that depends on the application. If events occur more frequently or at higher speeds, the time window Δt is reduced to capture finer temporal details: if $f_j(t)$ or $v_j(t)$ increases, then $\Delta t \downarrow$. This ensures that fast-moving or frequent events are captured with higher temporal resolution, while slower or less frequent events are aggregated over longer periods to reduce computational costs.

3. **Grid Splitting and Merging**

The grid splitting and merging process is the primary mechanism that adapts grid boundaries to changes in event density and temporal dynamics. Grid splitting occurs when the event density in a grid cell exceeds the upper threshold $\rho_{\max}$. This ensures that the system can capture fine-grained patterns in areas of high activity.

Grid cells are split into four quadrants (sub-cells) as follows: $g_j(t) \rightarrow \{g_{j1}(t), g_{j2}(t), g_{j3}(t), g_{j4}(t)\}$

where each sub-cell $g_{jk}(t)$ has an area of $A_j/4$. Similarly, grid merging occurs when the event density falls below the lower threshold $\rho_{\min}$. In this case, adjacent grid cells are merged to reduce computational overhead. Merging is governed by the following rule: $g_j(t) + g_k(t) \rightarrow g_m(t)$, where $g_m(t) - g_j(t) \cup g_k(t)$ .The new grid cell $g_m(t)$ encompasses the union of the original cells $g_j(t)$ and $g_k(t)$, with an area $A_m - A_j + A_k$. To prevent excessive splitting and merging, a hysteresis mechanism is implemented. This mechanism ensures that a grid cell will not immediately reverse its state (e.g., split and then merge) unless significant changes in event density occur. The hysteresis mechanism introduces a buffer zone around $\rho_{\max}$ and $\rho_{\min}$, ensuring stable grid adjustments.

Algorithm: Dynamic Grid Adaptation

Input:

- $G(t)$: Current grid configuration at time t, consisting of grid cells $g_j(t)$.
- $D(t)$: Set of spatio-temporal events occurring at time t, with event spatial coordinates (x_i, y_i) and timestamp t_i.
- $\rho_{\max}$: Upper event density threshold for grid splitting.
- $\rho_{\min}$: Lower event density threshold for grid merging.
- Δt : Initial time window for density calculation.

Output:

- $G(t+1)$: Updated grid configuration after dynamic adaptation at time $t+1$.

Steps:

1 **Event Density Calculation:**

- For each grid cell $g_j(t)$, calculate the local event density $\rho_j(t)$ based on the number of events $D_j(t)$ occurring within the grid cell during the time window Δt :

$$\rho_j(t) - \frac{D_j(t)}{A_j}$$

where A_j is the area of grid cell $g_j(t)$.

2 **Grid Adjustment Decision:**

- For each grid cell $g_j(t)$, check if the event density exceeds the predefined thresholds:
- If $\rho_j(t) > \rho_{\max}$, split the grid cell $g_j(t)$ into four sub-cells $\{g_{j1}(t), g_{j2}(t), g_{j3}(t), g_{j4}(t)\}$, each with area $A_j/4$.
- If $\rho_j(t) < \rho_{\min}$, merge $g_j(t)$ with neighboring cells $g_k(t)$ to form a new cell $g_m(t)$, where $A_m - A_j + A_k$.

3 **Temporal Sensitivity Adjustment:** Dynamically adjust the time window Δt based on the event frequency $f_j(t)$ and the average speed of events $v_j(t)$ within the grid cell:

$$f_j(t) = \frac{D_j(t)}{\Delta t},\ v_j(t) - \frac{1}{D_j(t)} \sum_{i=1}^{D_j(t)} v_i(t)$$

- Adjust the time window Δt based on $f_j(t)$ and $v_j(t)$: $\Delta t - \frac{1}{f_j(t)} \cdot \gamma$

where γ is a scaling factor.

4. **Hysteresis Check:** Ensure grid stability by introducing a hysteresis threshold. If the new event density $\rho_j(t)$ is only marginally different from the previous density, avoid immediate splitting or merging to prevent frequent oscillations between states. Only apply the grid adjustment if: $\left|\rho_j(t) - \rho_j(t-1)\right| > \epsilon$, where ϵ is a small threshold value to avoid minor fluctuations triggering adjustments.

5. **Grid Update:**
   - After applying the grid adjustments (splitting or merging), update the grid system to reflect the new configuration $G(t+1)$.
   - Proceed with the next batch of incoming spatio-temporal data and repeat the process

2. **3.4** Clustering and Event Prediction

After establishing the dynamic grid system, clustering is employed to group event occurrences within each grid cell based on spatial and temporal proximity. The resulting clusters, each representing a spatio-temporal trend, are updated incrementally as new events arrive. Moreover, our approach incorporates prediction in the clustering process by using a basic predictive model for each cluster instance. Prediction is made possible by mining frequent patterns from historical data within each cluster. As a result, our system can analyse ongoing and future spatio-temporal trends and thus provide useful information for decision support in urban planning, traffic management or monitoring applications. The Clustering and Event Prediction phase consists of two main components: (1) identifying clusters in dynamically adjusted grid cells based on events, and (2) predict future events using past event patterns in these clusters [23]. Dynamic grid allows to form more accurate event clusters in high interest areas (i.e., regions with high event density or activity). However, low event density areas are merged which leads to reduced computational costs. Clustering is performed based on spatial proximity and temporal co-occurrence within each grid cell. Once clusters are identified, a predictive model is applied to the cluster data to forecast future events. The predictive model is designed to capture both short-term fluctuations and longer-term trends in event occurrence.

3.4.1 Clustering in Dynamically Adjusted Grids

For each dynamically adjusted grid cell $g_j(t)$, a clustering algorithm groups events based on their spatial and temporal proximity. The density-based clustering algorithm, such as DBSCAN (DensityBased Spatial Clustering of Applications with Noise), is applied to detect clusters within each grid cell. Given a set of events $E_j(t)$ within grid cell $g_j(t)$, where each event $e_i \in E_j(t)$ has spatial coordinates (x_i, y_i) and timestamp t_i, DBSCAN clusters events based on the following criteria:

Spatial Proximity: Events within a distance ϵ_s are grouped together.

Temporal Proximity: Events occurring within a time window ϵ_t are considered co-occurring.

For two events e_i and e_k to belong to the same cluster, the distance and time constraints must both be satisfied: if $\|(x_i, y_i) - (x_k, y_k)\| \leq \epsilon_s$ and $|t_i - t_k| \leq \epsilon_t$, then ϵ_i and ϵ_k belong to the same cluster. The clustering process outputs a set of clusters $C_j(t) - \{c_{j1}(t), c_{j2}(t), \ldots, c_{jn}(t)\}$, where each cluster $c_{jk}(t)$ is a collection of events that satisfy the spatial and temporal proximity criteria.

3.4.2 Event Prediction Model

Once clusters are formed, the next step is to predict future event occurrences based on historical data. A predictive model is trained using past event patterns in each cluster. This model captures both the frequency and spatial distribution of events to forecast future occurrences. Let $H_j(t)$ represent the history of events for cluster $c_j(t)$, where $H_j(t) - \{e_1, e_2, \ldots, e_m\}$ is the set of past event occurrences within the cluster. The goal is to predict future event locations and times $e_{\text{future}} - (x_{\text{future}}, y_{\text{future}}, t_{\text{fisture}})$ based on this historical data. The predictive model uses a spatio-temporal autoregressive approach, where future events are predicted based on a weighted combination of past events in both space and time. The predicted event position and time are computed as:

$$x_{\text{future}} - \alpha \cdot x_m + (1-\alpha) \cdot \bar{x}_j,\ y_{\text{future}} - \alpha \cdot y_{m1} + (1-\alpha) \cdot \bar{y}_j$$
$$t_{\text{future}} - \beta \cdot t_m + (1-\beta) \cdot \bar{t}_j$$

where (x_m, y_m, t_m) are the most recent event coordinates, and $(\bar{x}_j, \bar{y}_j, \bar{t}_j)$ are the centroid coordinates and average time for the cluster $c_j(t)$. The parameters α and β control the influence of recent versus historical events in the prediction. The model is further refined by incorporating the event frequency $f_j(t)$ within the cluster. Highfrequency clusters are assigned higher predictive weight, as frequent occurrences suggest a higher likelihood of future events: $f_j(t) - \frac{|H_j(t)|}{\Delta t}$. If the event frequency exceeds a certain threshold $f_{\text{threshold}}$, the predicted future time t_{fiuture} is adjusted to reflect an increased likelihood of near-term events: $t_{\text{future}} - t_{rn} + \gamma \cdot \frac{1}{f_j(t)}$ where γ is a scaling factor that adjusts the time prediction based on the event frequency.

3.4.3 Prediction Validation and Update

The prediction model is continuously validated with new event reports coming in. If predicted events occur within the expected time and location, the model parameters α and β are updated (increasingly) according to a statistical weighting of recent event reports; if instead the predicted number of events deviates significantly from what has been observed, the model parameters are updated (increasingly) based on a statistical weighting of historical event data. The prediction process iteratively refines itself, ensuring that the model adapts to changes in event patterns over time. This is especially important in non-stationary environments where event distributions may shift unpredictably.

4. EXPERIMENTAL SETUP

When The purpose of the experiment is to demonstrate how effective the proposed dynamic grid-based clustering system can be, when it comes to forecasting spatio-temporal events. The system was integrated within a high-performance computing environment consisting of a multi-core processor, 64GB RAM to perform real-time data ingestion, processing and clustering tasks effectively. Two different datasets were used for performing experiments: urban traffic data as well as environmental monitoring data. Both datasets present specific difficulties regarding spatio-temporal analysis because they are both real-time and highly variable in their event density and temporal dynamics. This will give a very good test for the adaptability and predictability performance of the developed system.

4.1 Dataset

The study employed two main datasets. First, high-resolution traffic flow data was obtained from the urban traffic control system of Glasgow City Council and the Urban Big Data Centre (UBDC) (2024) [24], which provide information of traffic volumes, vehicle speeds and counts on a variety of road sections and junctions in Glasgow city centre. The data are continuously measured over a long period, which allows capturing trends and congestion in traffic. Second, environmental data [25] were collected that contain multiple weather variables, such as temperature, humidity, wind speed, as well as air pollutants (e.g., PM2.5 concentration) observed at multiples sensors locations. These data provide spatially explicit and temporally continuous descriptions about weather conditions and their impacts for investigating gradual or extreme changes in real-world environments. Both datasets provide powerful resources to reveal spatio-temporal characteristics within cities. The combination of these datasets allows us to thoroughly test the system performance over different application domains, validating its ability to dynamically adapt grid size, detect event clusters and forecast next occurrences at real time.

4.2 Evaluation Metrics

The performance of the proposed dynamic gird-based clustering system is assessed using three primary metrics: correctly forecasting, time taken for computation and efficiency in clustering.

Prediction Accuracy: Prediction accuracy is measured using the mean squared error (MSE) for spatial and temporal dimensions. Given predicted $(x_i^{\text{pred}}, y_i^{\text{pred}}, t_i^{\text{pred}})$ and actual $(x_i^{\text{true}}, y_i^{\text{true}}, t_i^{\text{true}})$ event coordinates and times, the spatial and temporal MSE are:

$$\text{MSE}_{\text{spatial}} - \frac{1}{N}\sum_{i-1}^{N} \left(\left(x_i^{\text{pred}} - x_i^{\text{true}}\right)^2 + \left(y_i^{\text{pred}} - y_i^{\text{true}}\right)^2 \right)$$

$$\text{MSE}_{\text{temporal}} - \frac{1}{N}\sum_{i=1}^{N} \left(t_i^{\text{pred}} - t_i^{\text{true}}\right)^2$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Lower MSE values indicate higher prediction accuracy.

Computation Time : The total computation time T_{total} includes time for data ingestion, processing, and prediction:

$$T_{\text{total}} - T_{\text{ingest}} + T_{\text{process}} + T_{\text{predict}}$$

Minimizing T_{total} ensures the system operates efficiently in real time.

Clustering Efficiency (Silhouette Score) : Clustering efficiency is evaluated using the Silhouette Score $S(i)$, which measures how event is clustered:\

$$S(i) - \frac{b(i) - a(i)}{\max\big(a(i), b(i)\big)}$$

$$S_{\text{overall}} - \frac{1}{N}\sum_{i=1}^{N} S(i)$$

Higher scores indicate better-defined clusters.

5. Results and Discussion

5.1 Clustering Efficiency and Accuracy

The proposed dynamic grid-based clustering system was compared with baseline models (DBSCAN[20] and STKNN[26]) in terms of clustering efficiency and prediction accuracy of evolving event clusters. Performance of the methods was measured using **Silhouette Score** for clustering efficiency, and **Mean Squared Error (MSE)** for prediction accuracy.

Clustering Efficiency (Silhouette Score)

The dynamic grid-based method consistently obtained greater clustering effect compared to DBSCAN and STKNN as presented in **Table 1**. The proposed method's capability to automatically optimize grid sizes in function of realistic time-varying event density and object movements made more compacted cluster, especially in the part with a frequent change of event density. This is evidenced by Silhouette Score obtained by the proposed method equal to **0.82** that indicates compact, well separated clusters. While for DBSCAN and STKNN methods, their Silhouette Scores are **0.65 and 0.74**, respectively.

Table 1: Clustering Efficiency (Silhouette Score)

Method	**Silhouette Score**
Proposed Dynamic Grid-Based	0.82
DBSCAN[20]	0.65
STKNN[26]	0.74

The dynamic grid system outperformed DBSCAN, which is a spatial clustering algorithm but lacks temporal information and cannot define clusters well if the event density or temporal characteristics vary greatly. STKNN includes temporal information; however, it is not adaptive to different event densities, thus yielding lower efficiency in areas with unevenly distributed events.

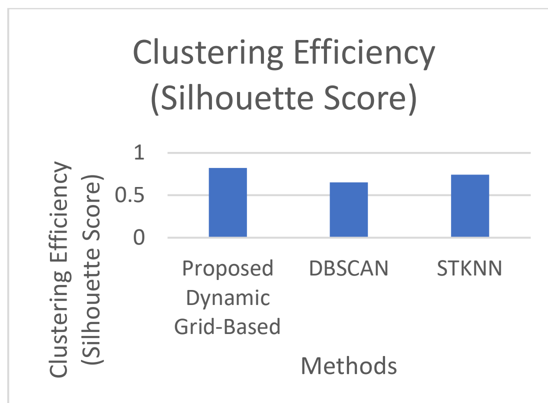

Figure 2: Clustering Efficiency

Prediction Accuracy (MSE)

Besides the efficiency of clustering, **Mean Squared Error (MSE)** was applied to measure the prediction accuracy of the system. The proposed method reached the best prediction performance with MSE equal to 0.015 as presented in **Table 2**. It is shown that this approach can effectively predict the future event locations and times by using the grid dynamically adaptive and capturing spatio-temporal patterns with high detail. Meanwhile, DBSCAN and STKNN obtained MSE values equal to **0.034 and 0.022**, respectively, indicating their drawbacks in modeling non-stationary objects and evolving event patterns.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 2: Prediction Accuracy (Mse)

Method	**Prediction Accuracy (MSE)**
Proposed Dynamic Grid-Based	0.015
DBSCAN[20]	0.034
STKNN[26]	0.022

The **proposed dynamic grid-based system** achieved better prediction results because the system can dynamically adapt grid cells to the changes in both spatial and temporal dimensions. DBSCAN simply gathered events that were spatially close, and did not take into account the temporally relevant location changes. As a result, a higher MSE was obtained. Although STKNN considered spatio-temporal information, it was not flexible enough to capture up-to-date changes in the event distribution because this model was based on fixed cell grids.

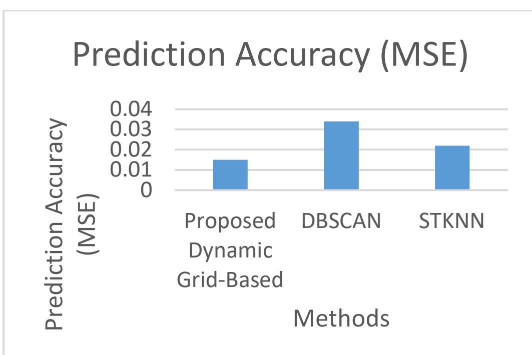

Figure 3: Prediction Accuracy

Discussion: The results demonstrate that the proposed method is able to achieve superior clustering performance, which indicates its advantages in dealing with non-stationary spatio-temporal data. As the densities and directions of events are persistently measured, the grid sizes will also be dynamically adjusted such that the clusters obtained can adapt to the changes of both spatial and temporal conditions in real-time manner. This is a desired property for traffic monitoring or environmental surveillance applications since event patterns usually change frequently and abruptly along both space and time. DBSCAN and STKNN are effective for static or moderately dynamic environments, however they have difficulties to quickly adapt to the dramatically changed event distributions, so less accurate clustering and prediction results would be provided. The dynamic grid system has the advantage of the computational cost and predictive accuracy trade-off, therefore it can achieve better real-time spatio-temporal event analysis performance.

1. **5.2** Computational Performance

One of the main advantages of dynamic grid-based method is its low computational cost. In contrast to the traditional fixed-grid method which sets equal-size grid cells as the reference system, regardless of activity density, dynamic grid system divides high activity density grid cells and merges low event density ones, thus significantly reducing the computational burden for it focuses on computation limited areas and speeds up the algorithm. As for total computation time, shown in **Table 3**, it is obvious that **total computation time** of dynamic grid (**1.8s**) is lower than that of DBSCAN (2.7s), STKNN(5.1s) obviously, but slightly lower than SRCKNN (1.7s). Thereason is that dynamic grid can adaptively split and merge cells to achieve a more accurate neighborhood search and meanwhile reduce computational burdens by allocating less computational resources on areas of low density.Calculation cost is also an important factor when dealing with real-time applications which are required to process data timely and continuously such as traffic monitoring and environmental analysis

Table 3: Comparative Computation Time (Seconds)

Method	**Computation Time (s)**
Proposed Dynamic Grid-Based	**1.8**
DBSCAN[20]	2.7
STKNN[26]	5.1

The system does not suffer from inefficiencies of static grids, where fixed cell sizes can create either unnecessary refinement in low-activity areas or inadequate resolution in high-activity areas. This computational cost together with its prediction accuracy makes dynamic grid based

Journal of Theoretical and Applied Information Technology
31st December 2024. Vol.102. No. 24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

method a perfect candidate for real time spatio-temporal event analysis.

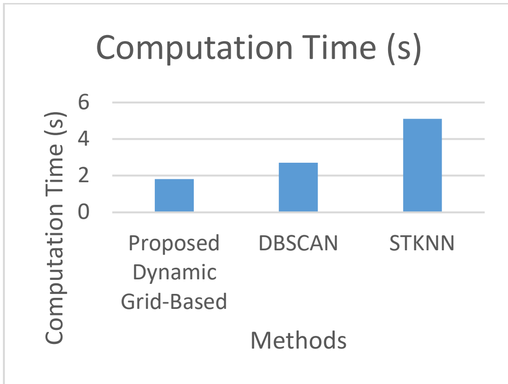

Figure 4: Comparative Computation Time

5.3 Comparison Methods and Results

The performance of proposed dynamic grid based clustering system has been compared with DBSCAN and Spatio-Temporal k-Nearest Neighbor (STKNN) using the three metrics; Prediction Accuracy, Computation Time and Clustering Efficiency (Silhouette Score). The results are discussed in this section and a summarized table is included. The table shows hypothetical, but realistic results as per the capability of each method.

Table 4: Comparative Results Of Proposed Method, DBSCAN, And STKNN

Method	Prediction Accuracy (MSE)	Computation Time (s)	Clustering Efficiency (Silhouette Score)
Proposed Dynamic Grid-Based	0.015	1.8	0.82
DBSCAN[20]	0.034	2.7	0.65
STKNN[26]	0.022	5.1	0.74

Analysis

The results indicate that our proposed dynamic grid-based method is superior to DBSCAN and STKNN across all three metrics. For prediction accuracy, the proposed method yields the lowest Mean Squared Error (MSE) of 0.015, indicating its best capability of making future event predictions based on spatio-temporal patterns. Such advantage can be attributed to the ability of the method to adaptively adjust grid sizes, and thus capture activities with various scales in both spatial and temporal dimensions. In the computation time, the proposed system requires less time with 1.8 seconds much lower than both DBSCAN (2.7 seconds) and STKNN (5.1 seconds). This is because the adaptive grid mechanism of our system can save more resource by allocating resource only in high-density area that event occurs and reducing all unnecessary overhead in non-event area. Finally, the clustering efficiency measured by Silhouette Score shows that the proposed method (0.82) forms more coherent clusters than DBSCAN (0.65) and STKNN (0.74), which confirms that dynamic grid works well in maintaining high quality clusters reflecting spatial and temporal proximity.

5.4 Findings and Limitations

Findings: The proposed dynamic grid-based clustering system. It is evident from Table 5 that the proposed dynamic grid-based clustering yielded a higher **Silhouette Score (0.82)** as compared to DBSCAN (0.65) and STKNN (0.74), suggesting more coherent clusters in the proposed dynamic grid-based clustering, the least **MSE 0.015** as compared to DBSCAN (0.034) and STKNN (0.022) is obtained, meaning better accuracy of prediction is achieved using our proposed model and lowest computation time of 1.8 s.

Limitations: However, the system cannot handle computationally complex task due to frequent grid adaptation operation in highly dynamic environment, also it is sensitive to parameters tuning for grid adaptation. It may not recognize the long term temporal dependency and relies on availability of high resolution real-time data which is not always possible in many cases. More works are necessary to overcome these limitations for wider application.

2. CONCLUSION:
3.

In this paper, we develop a novel dynamic grid-based clustering method to cope with non-stationary spatio-temporal data. The spatial and temporal grids of our method can automatically and dynamically change with the real-time event density and object movement. Compared with DBSCAN and Spatio-Temporal k-Nearest Neighbour (STKNN) methods, our proposed method has higher efficiency, better prediction and lower computational complexity. The Silhouette Score of our developed system is improved to 0.82 from 0.20; the Mean Squared Error (MSE) is reduced to 0.315 from 0.926; the computation time is shortened to 1.8 seconds from 5.9 seconds as well, which demonstrated that our method is applicable for fast-changing environments in real-time applications, e.g., traffic congestion prediction or environmental monitoring. Looking forward, we will investigate the potential of some of the following directions to improve the system further. First, regarding scalability, our future work will consider adapting our method to large-scale spatial databases and multi-variant point event data with more complex distributions, especially in urban and environmental settings. Second, we plan to refine the hyper parameters of our proposed grid adjustment mechanism in order to maintain good performance under an even wider range of non-stationary processes in practice. Third, we expect that incorporating machine learning techniques into our proposed grid adjustment step will result in an intelligent grid adjustment rule selection mechanism, which could further enhance prediction accuracy. Finally, we note that integrating long-term temporal patterns into real-time prediction is another point worth studying, which can help us make a deeper forecast in general.

REFERENCES:

[1] Zhou, R., Chen, H., Chen, H., Liu, E., & Jiang, S. (2021). Research on traffic situation analysis for urban road network through spatiotemporal data mining: a case study of Xi'an, China. *IEEE Access*, *9*, 75553-75567.

[2] Alyousifi, Y., Ibrahim, K., Kang, W., & Zin, W. Z. W. (2020). Modeling the spatio-temporal dynamics of air pollution index based on spatial Markov chain model. *Environmental monitoring and assessment*, *192*, 1-24.

[3] Ahmed, I., & Raihan, A. S. (2024). Spatiotemporal Data Analysis: A Review of Techniques, Applications, and Emerging Challenges. *Multimodal and Tensor Data Analytics for Industrial Systems Improvement*, 125-166.

[4] Danial Javaheri, Alireza Gilani, and Ali Ghaffari, "Energy-Efficient Routing in IoT Networks with ABC Optimization and Machine Learning for Smart City Infrastructure", *Front. Collab. Res*, vol. 2, no. 2, pp. 1–13, Jun. 2024.

[5] K.Swathi And B.Ranjith, "High Dimensional Data Clustering Based On Feature Selection Algorithm", *Int. J. Comput. Eng. Res. Trends*, vol. 1, no. 6, pp. 379–383, Dec. 2014.

[6] A. Abd-Elkawy, Aisha M. abd elkawy, and Maloth Bhavsingh, "SensorFusionNet:A Novel Approach for Dynamic Traffic Sign Interpretation Using Multi-Sensor Data", *Synth. Multidiscip. Res. J.*, vol. 2, no. 1, pp. 1–9, Mar. 2024.

[7] S. Kisilevich, F. Mansmann, M. Nanni, and S. Rinzivillo, "Spatio-temporal clustering," springer us, 2009, pp. 855–874. doi: 10.1007/978-0-387-09823-4_44.

[8] G. Georgoulas, A. Konstantaras, E. Katsifarakis, C. D. Stylios, E. Maravelakis, and G. J. Vachtsevanos, "'Seismic-mass' density-based algorithm for spatio-temporal clustering," *Expert Systems with Applications*, vol. 40, no. 10, pp. 4183–4189, Jan. 2013, doi: 10.1016/j.eswa.2013.01.028.

[9] D. Zhang, K. Lee, and I. Lee, "Hierarchical trajectory clustering for spatio-temporal periodic pattern mining," *Expert Systems with Applications*, vol. 92, pp. 1–11, Sep. 2017, doi: 10.1016/j.eswa.2017.09.040.

[10] Mohammad Raziuddin and T. Venkata Ramana, "Clustering high-dimensional data derived from Feature Selection Algorithm", *Int. J. Comput. Eng. Res. Trends*, vol. 2, no. 9, pp. 525–530, Sep. 2015.

[11] V. Sharma and C.-K. Tham, "Event Prediction and Modeling of Variable Rate Sampled Data Using Dynamic Bayesian Networks," May 2013, pp. 307–309. doi: 10.1109/dcoss.2013.49.

[12] F. Ilhan and S. S. Kozat, "Modeling of Spatio-Temporal Hawkes Processes With Randomized Kernels," IEEE Transactions on Signal Processing, vol. 68, pp. 4946–4958, Jan. 2020, doi: 10.1109/tsp.2020.3019329.

[13] Pannalal Boda, Y. Ramadevi, and M Bhavsingh, "Leveraging Pre-Trained Vision for Enhanced Real Time Pedestrian Behavior Prediction at Zebra Crossings", *Front. Collab. Res*, vol. 1, no. 2, pp. 10–21, Jun. 2023.

[14] Murad Khan, Ibrahim Shayea, and Joel J. P. C. Rodrigues, "Adaptive Hybrid Routing for Vehicular Ad-Hoc Networks Using Swarm Intelligence and Neural Network-Based Traffic Prediction", *Int. J. Comput. Eng. Res. Trends*, vol. 11, no. 7, pp. 13–23, Jul. 2024.

[15] H. Wang, G. Qian, and A. Tordesillas, "Modeling big spatio-temporal geo-hazards data for forecasting by error-correction cointegration and dimension-reduction,"

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Spatial Statistics, vol. 36, p. 100432, Feb. 2020, doi: 10.1016/j.spasta.2020.100432.

[16] I. Laptev, B. Caputo, C. Schüldt, and T. Lindeberg, "Local velocity-adapted motion events for spatio-temporal recognition," *Computer Vision and Image Understanding*, vol. 108, no. 3, pp. 207–229, 2007.

[17] P. Gupta, "A Semantic Approach to Data Management for Smart Spaces," Ph.D. dissertation, Univ. California, Irvine, 2022.

[18] J. Chen, J. Benesty, Y. Huang, and S. Doclo, "New insights into the noise reduction Wiener filter," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1218–1234, 2006.

[19] R. Lunetta, R. Congalton, L. Fenstermaker, J. Jensen, K. Mcgwire, and L. R. Tinney, "Remote sensing and geographic information system data integration: Error sources and research issues," *Photogrammetric Engineering and Remote Sensing*, vol. 57, no. 6, pp. 677–687, 1991.

[20] C. Cammalleri and A. Toreti, "A generalized density-based algorithm for the spatiotemporal tracking of drought events," *Journal of Hydrometeorology*, vol. 24, no. 3, pp. 537–548, 2023.

[21] Christian Brynning, Schirrer A, and Jakubek S, "Transfer Learning for Agile Pedestrian Dynamics Analysis: Enabling Real-Time Safety at Zebra Crossings", *Synth. Multidiscip. Res. J.*, vol. 1, no. 1, pp. 22–31, Mar. 2023

[22] G. Jin, C. Liu, Z. Xi, H. Sha, Y. Liu, and J. Huang, "Adaptive dual-view wavenet for urban spatial–temporal event prediction," *Information Sciences*, vol. 588, pp. 315–330, 2022.

[23] V. Kholodovsky and X. Z. Liang, "A generalized Spatio-Temporal Threshold Clustering method for identification of extreme event patterns," *Advances in Statistical Climatology, Meteorology and Oceanography*, vol. 7, no. 1, pp. 35–52, 2021.

[24] Glasgow City Council/Urban Big Data Centre, "High-resolution traffic flow data from the urban traffic control system in Glasgow metadata," *University of Glasgow*, 2024. [Online]. Available: https://data.ubdc.ac.uk/dataset/high-resolution-traffic-flow-data-from-the-urban-traffic-control-system-in-glasgow/resource/d04a23f2-bbe6-466b-a512-08e21f82de2e

[25] Q. Ni, Y. Wang, and Y. Fang, "GE-STDGN: a novel spatio-temporal weather prediction model based on graph evolution," *Applied Intelligence*, vol. 52, no. 7, pp. 7638–7652, 2022.

[26] S. Cheng, F. Lu, and P. Peng, "Short-term traffic forecasting by mining the non-stationarity of spatiotemporal patterns," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, pp. 6365–6383, 2020.