# DETECTION OF BEHAVIOURAL BASELINE DEVIATION IN ENDPOINT USAGE THROUGH MOUSE DYNAMICS ANALYSIS

*Kamran N. Asgarov**

Department of Engineering Mathematics and Artificial Intelligence, Azerbaijan Technical University, Baku
Azerbaijan

* Corresponding Author, e-mail: kamran.asgarov.n@student.aztu.edu.az

**Abstract:** This paper introduces a two-stage framework for detecting deviations in user behaviour on endpoints via mouse-movement analysis. In the training phase, raw cursor telemetry is converted into minute-level heatmaps, reduced in dimensionality and subjected to a systematic evaluation of non-parametric clustering algorithms to extract representative core samples of distinct operational states. In the detection phase, multiple comparison engines—including distance-thresholding, margin-based novelty detection and ensemble isolation—were rigorously benchmarked against synthetic heatmap sequences with known anomaly labels. A single clustering engine and a single comparison method, chosen for their superior trade-off between sensitivity and precision, were integrated into the final pipeline. This methodology delivers robust, interpretable, real-time identification of behavioural baseline deviations, thereby enhancing endpoint security through user-centric anomaly detection.

**Key words:** endpoint security, anomaly detection, user behaviour, baseline deviation, mouse dynamics.

## 1. INTRODUCTION

Numerous programs and techniques have been developed to protect endpoints. These include anti-virus (AV) programs that primarily perform static analysis using signature-based detection methods to identify malware of different categories [1], anti-ransomware platforms (ARPs) that focus exclusively on detecting one category of malware – ransomwares [2], and endpoint detection and response (EDR) systems that provide end-to-end security by combining multiple detection and monitoring methods across multiple endpoints in networks, such as in an organisations. Each of these programs has its own area of protection and monitoring with specific tools.

Given the vast and diverse infrastructure of contemporary endpoint deployments, each group of endpoints often requires a tailored set of security and protection tools.

Consequently, a wide array of endpoint security programs exists, each designed to secure and monitor specific aspects of endpoint activity. While these programs may address different security needs, many share common analytical tools and techniques. For example, process monitoring, event monitoring, known malware database comparisons are common features for those programs. One technique, such as dynamic analysis could be critically important for certain security solutions, but less so for others [3]. Among the array of security tools, behaviour-based analysis methods stand out due to their unique emphasis on monitoring user interactions with endpoints.

One tool that is often used in endpoint security software is behavioural analysis, which can include analysis of the behaviour of programs and processes, or telemetry metrics of endpoints [4], or analysis of user behaviour when managing an endpoint. This paper focuses particularly on the behavioural analysis of user interactions with endpoints. User behaviour analysis can leverage diverse data sources, including the types of applications utilised, duration of application usage, active time intervals, network connections, visited websites, and patterns of interaction with input devices. Such behavioural indicators include typing speed, mouse cursor movements, and navigation patterns. By examining these behavioural dynamics, particularly mouse movement patterns, this study aims to identify baseline deviations that may indicate anomalies or potential security threats.

A recent study in a different tier of the security stack further underscores the promise of baseline-deviation analytics: Article [5] shows that Relying Parties in OpenID Connect federations can construct lightweight session fingerprints from protocol-level attributes and, by monitoring their drift, detect credential theft and session hijacking in real time. Their findings confirm that subtle behavioural shifts, whether captured in HTTP exchanges or low-level input-device telemetry, provide an effective early-warning signal, and they motivate our decision to pursue endpoint-resident mouse-dynamics modelling rather than rely solely on higher-layer metadata.

Unlike prior studies that apply mouse dynamics either to authenticate the current user or to visualise attention, our work treats continuous cursor telemetry as a problem of deviating the behavioural baseline for endpoint security. Specifically, we present a minute-level heatmap that compresses raw motion, click, and scroll events while preserving spatiotemporal context; construct this baseline without any a priori assumptions about the number of operating states, relying solely on non-parametric density-based clustering; and combine the resulting core samples with lightweight single-class comparison mechanisms to provide interpretable real-time anomaly flags suitable for fenced endpoints. Collectively, these advances extend mouse dynamics research beyond identity verification and lay the foundation for low-cost, user-centric threat detection on everyday endpoints.

## 2. RELATED WORKS

The analysis of mouse dynamics has attracted considerable interest across various research areas, particularly in user authentication, user attention measurement, and behavioural biometrics.

Several researchers have investigated mouse dynamics for user authentication and behavioural analysis. Deep learning approaches and hybrid authentication methods, respectively, to leverage mouse dynamics and keystroke dynamics for user authentication are discussed in [6, 7]. A method for assessing user behaviour through mouse movements by treating user identification as a one-class classification problem is proposed in [8]. Their preliminary results suggested that individual behavioural models, despite varying effectiveness among users, offer promising directions for behavioural biometrics research.

In the context of user attention measurement, article [9] proposes Word Attention Heatmaps, using mouse movement heatmaps to visually represent user attention to textual content, thereby aiding content improvement and readability enhancement. In addition, the need for standardization and methodological rigor in mouse dynamics research is emphasized in [10], revealing hardware and configuration differences significantly impact the reliability of mouse metrics [10].

In contrast to earlier work on authentication [6, 7], biometrics [8, 11] and attention visualisation [8], our study frames mouse telemetry as a baseline-deviation problem for security. We model minute-level heat-maps, form a multi-cluster baseline with OPTICS/HDBSCAN, and apply one-class detectors to spot anomalies—a scenario largely overlooked in the existing mouse-dynamics literature.

## 3. THE MAIN PART

Most endpoint interactions by users involve the use of cursor control devices such as a mouse or trackpad. These input devices play a crucial role in operating system interactions, thereby capturing significant aspects of user behaviour.

### 3.1. Mouse dynamics data collection and representation

Operating systems record extensive telemetry data related to mouse interactions, including cursor coordinates, click events, scroll vectors, and associated timestamps. Formally, mouse telemetry data can be represented as a multivariate time-series vector $m_t = (l, x, y, l, r, s_x, s_y)$, where $t$ denotes the timestamp, $x, y$ are cursor coordinates, $l, r$ represent binary indicators of left and right clicks, respectively, and $s_x, s_y$ indicate scroll direction vectors. Although the temporal nature of such data is critical for accurate baseline construction, the complexity of multivariate time-series data presents analytical challenges [12]. For example, Figure 1 illustrates the dynamics of mouse cursor which includes movement, click and scroll data over a period of 5 minutes. The figure shows the mouse cursor movement trajectory in blue curve, clicks on screen in red circles and scrolls with their directions in green arrows. Despite the small period, the cursor dynamics data is very large and chaotic, which complicates its analysis and the task of constructing a baseline for these dynamics.

To simplify the analysis, we turn the raw telemetry data into heatmaps that visualise the intensity and frequency of mouse interactions such as movements, clicks and cursor pointing. We generate heatmaps based on the collected mouse dynamics telemetry periodically at regular intervals. Thus, by varying the structure of the data, from a complex set of time series vectors, we obtain matrices where each element represents a

pixel on the endpoint monitor, with values scaled between 0 (minimum activity) and 1 (maximum activity).

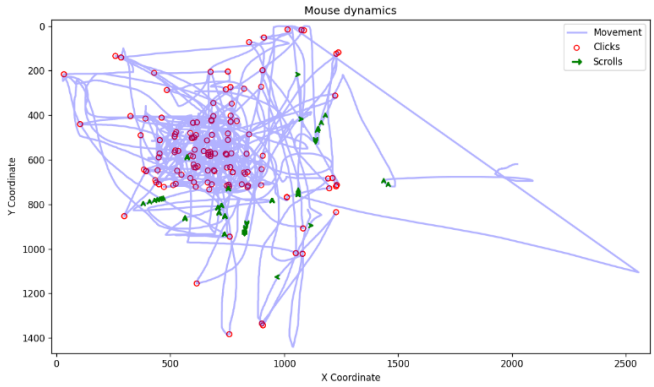


*Figure 1. Mouse cursor dynamics.*

In Figure 2, which shows an example heatmap of the mouse cursor dynamics data from Figure 1, the active and inactive regions of the cursor can be seen. This data structure does not aggregate all available cursor dynamics information, but it contains the most relevant information in the form of matrices and is applicable for many machine learning tools. In addition, periodic collection of mouse cursor dynamics data at small intervals allows to preserve time series information.

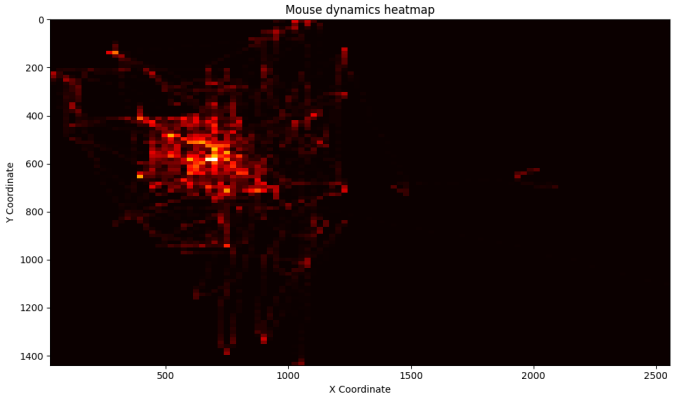


*Figure 2. Heatmap representation of mouse cursor dynamics.*

### 3.2. Dimensionality reduction and behavioural clustering

However, the high dimensionality of these matrices, corresponding directly to monitor resolution, complicates further analysis. To manage dimensional complexity, dimensionality reduction techniques such as convolution are employed [13]. Figure 3 illustrates the changes of the heatmap when the dimensionality of the matrix is reduced. This process simplifies each heatmap matrix into a lower-dimensional representation, subsequently transformed into vectors by concatenating matrix rows.

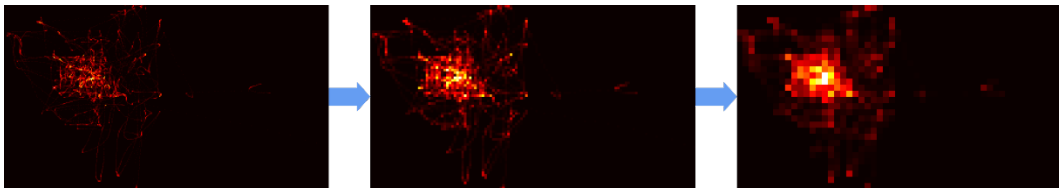

*Figure 3. Heatmap dimension reduction.*

As a result, we have transformed the original mouse dynamics data structure $m_t = (l, x, y, l, r, s_x, s_y)$ into a set of vectors $V_{[t_i,t_{i+1}]} = \{(v_1, \dots, v_N) \mid v_k \in [0,1]\}$ whose values vary between 0 and 1. Here, the value of $N$ is a hyperparameter of this dimensionality reduction approach and depends on how much the dimensionality of the matrices is reduced. The set of these vectors will constitute our baseline of the user's endpoint control behaviour based on mouse cursor dynamics $\mathcal{B} = \{V_T \mid T \subset \mathcal{T}\}$ where $\mathcal{T}$ is training period. The baseline is constructed during the training period as follows, the software continuously listens and collects all mouse control events and every few minutes generates matrices based on these events that correspond to heat maps of cursor activity. Then the dimensionality of these matrices is reduced using reduction techniques to vectors that are stored in the baseline set.

Users commonly perform routine tasks, particularly in corporate environments such as banking or call centres, involving repetitive interactions with a limited set of software applications. This repetitive nature typically results in distinct operational states, identifiable as separate classes of cursor activity. During the training phase for baseline construction, mouse activity heatmaps are collected and analysed to identify the distinct user behaviour states. These identified states are represented as clusters within the vector set, $\mathcal{C} = \{C_1, C_2, \dots, C_n\}$, where each cluster corresponds to a specific behavioural state. The final set of these clusters constitutes the behavioural baseline $\mathcal{B}$.

### 3.3. Anomaly detection methodology

During the anomaly detection pipeline, we compare new telemetry vectors, obtained from reduced-dimensional heatmaps against the established baseline clusters. If a new vector aligns closely with any baseline cluster, the corresponding user behaviour is classified as normal. Conversely, if the vector does not match any known baseline cluster, it is categorised as anomaly, signalling potential security threats or behavioural deviations.

Methodology of anomaly detection program for user's mouse behavioural dynamics is illustrated in Figure 4. It consists of two main phases - training and monitoring (detection). In both phases, raw mouse-movement events are first captured by a dedicated metrics provider, which aggregates cursor coordinates, timestamps and contextual metadata in real time. These events are then transformed into minute-level heatmaps via a two-dimensional binning procedure, producing normalised intensity arrays that characterise usage patterns.

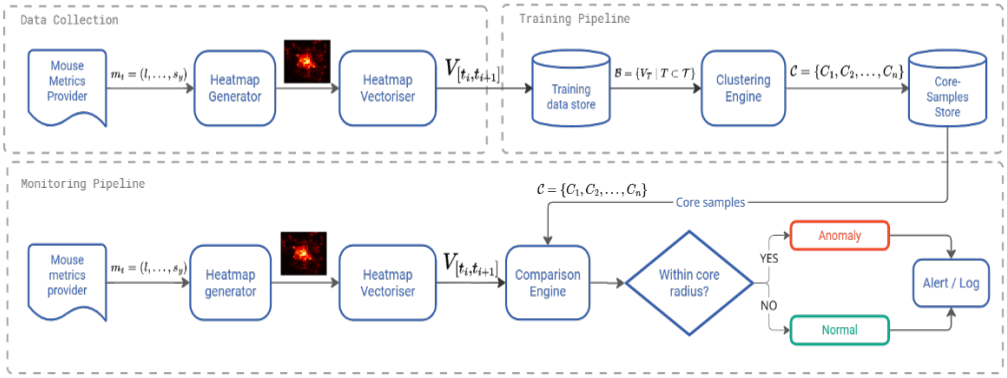


*Figure 4. Methodology pipelines flowchart.*

During an initial training phase, these heatmaps are stored and subjected to a clustering algorithm whose principal output comprises both cluster assignments and a subset of "core" samples that lie at the heart of each dense region. It is important to note that the clustering algorithm will have no information about the number or types of clusters when working with the data, as we have no information about user behaviour in advance. Therefore, methods such as k-nearest neighbours (KNN), K-medoids and Gaussian mixture models are not applicable in our problem. The only applicable clustering methods for our problem are non-parametric clustering methods. Unlike the "parametric" approaches described above, nonparametric methods allow the data itself to dictate how many groups exist.

These core samples that are obtained at the output of the clustering algorithm represent the most typical observations within each cluster-points lying in the centre of high-density regions and thus encapsulating the basic structure of each behavioural mode. Once extracted, they are passed to a dedicated comparison algorithm, which serves as the decision-making module during the detection phase. On detection phase each new mouse dynamics heatmap is evaluated against the stored core samples using a generalised similarity or divergence criterion: if the embedding falls within the acceptable limits of similarity to any baseline sample, it is considered to be consistent with the established baseline and is labelled as normal; if not, it is labelled as anomaly.

The proposed workflow is inherently stream-oriented and therefore suitable for real-time analysis on endpoints. During an initial training window $T_{tr}$, the resident agent captures raw cursor telemetry, converts each $\Delta t$–minute segment into a reduced-dimensional heat-map vector and stores the resulting sequence $V = \{V_1, \ldots, V_k\}$. When $T_{tr}$ elapses, the entire collection V is processed by a density-based, non-parametric clustering routine, the output of which is a compact set of core samples. The agent then transitions to the monitoring phase. Incoming mouse events are buffered, converted into fresh heat-maps, embedded in the same vector space, and compared against the core-sample library in a streaming loop. If the minimum divergence to any prototype is below an adaptive threshold $\theta$ of comparison method, the observation is labelled normal; otherwise, it is flagged anomalous, and an alert is issued. All of the above processing flow is illustrated in Figure 5, the system-architecture diagram.

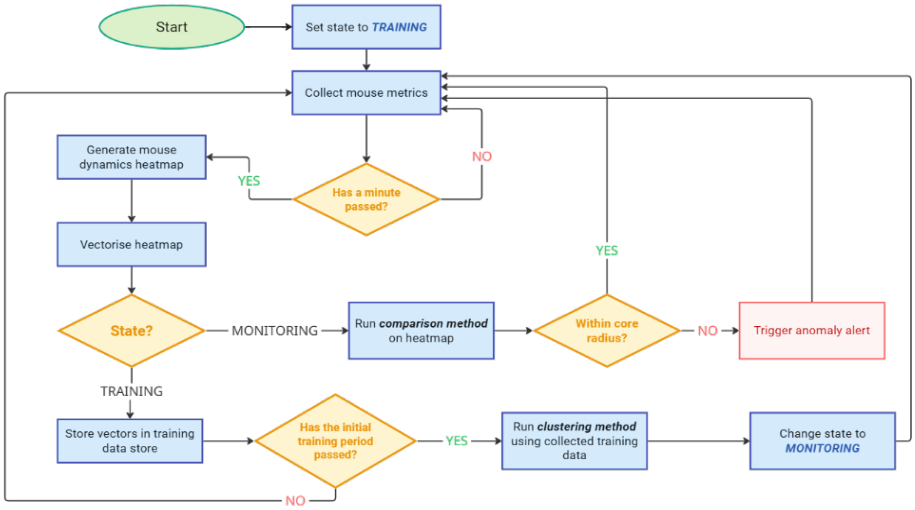


*Figure 5. System architecture diagram.*

### 3.4. Clustering methods

**DBSCAN.** Density-Based Spatial Clustering of Applications with Noise identifies core points that have at least minPts neighbours within radius $\varepsilon$ and links them by density reachability, producing arbitrarily shaped clusters while discarding sparse noise [14].

**OPTICS.** Ordering Points to Identify the Clustering Structure is a method that orders points by their reachability distance, allowing clusters to be extracted at any density threshold; pronounced valleys in the plot mark dense regions, while $\xi \in (0,1)$ and minClusterSize control their persistence [15].

**HDBSCAN.** Hierarchical Density-Based Spatial Clustering of Applications with Noise converts the OPTICS ordering into a condensed density tree, then selects the branches that maximise cluster stability, yielding a single, flat partition that is robust to parameter choice [16].

**BIRCH.** Balanced Iterative Reducing and Clustering using Hierarchies maintains a tree of compact summaries (CF-nodes) and assimilates new points on-line; only when a leaf's diameter exceeds $T$ is the node split, making the method memory-efficient for high-volume telemetry [17].

### 3.5. Comparison methods

In the detection phase, the core-sample set $\mathcal{C} = \{c_i\}$ derived during training is supplied to a comparison engine that evaluates each new embedding $x$. If $x$ does not sufficiently conform to any member of $\mathcal{C}$, it is flagged as anomalous. We investigate three distinct comparison paradigms-distance-based, margin-based and tree-based-as follows.

**Nearest neighbour.** Under this approach, anomaly detection reduces to a simple radius test. If the minimum Euclidean distance $r(x) \leq \varepsilon$, the heat-map is novel. This simple rule offers constant-time scoring once cores are indexed.

**One-class support vector machine.** This margin-based method fits a hypersphere (or hyperplane in feature space) around the core samples in novelty-detection mode [18]. A kernel boundary encloses the cores; its tightness is tuned by $\nu$. Points outside the boundary receive a negative decision value and are flagged.

**Isolation forest.** Random binary trees recursively partition the feature space [19]; genuine anomalies require fewer splits to isolate and thus obtain a higher anomaly score $s(x)$.

## 4. EXPERIMENTS AND RESULTS

In this section, we conduct experimental tests of the performance of the previously described methodology for detecting the deviation of a user's baseline behaviour based on mouse control metrics. For this purpose, clustering methods are initially tested and evaluated using the clustering algorithms' evaluation metrics. Then on the basis of the core clusters, which are obtained from the clustering method with the best results, the comparison methods are tested. All source code, synthetic datasets and configuration files required to replicate every experiment are available in the author's open-access repository [20]. The archive contains the telemetry recorder that logs timestamped cursor events, the renderer that produces the trajectory illustration reproduced in Figure 1, the heat-map generator responsible for the intensity plots in Figures 2 and 3, a simulator that fabricates state-labelled cursor streams for controlled benchmarking, and the notebooks that carry out the clustering grid-search and anomaly-detection trials reported in Tables 1 and 2.

Due to the absence of publicly available datasets comprising extended mouse dynamics with distinct operational states, we have generated synthetic mouse dynamics data through a simulation algorithm designed to emulate realistic cursor behaviours under various operational states. This simulation produces mouse movement heatmaps representative of different user interactions, such as typical browsing, detailed design work, and rapid navigation, by manipulating parameters such as trajectory length, movement variance, drift probabilities, and confined interaction regions. The simulated dataset comprises multiple distinct behavioural states, intentionally structured to include specific states exclusive to either the training or detection phases, thereby ensuring the robustness of anomaly detection evaluation. To emulate adversarial behaviour, we expose the framework to only four of the six cursor states (A, B, C, E) during training and withhold two states (D and F) until the detection phase [20]. Each heat-map generated under the unseen states is assigned a "malicious'' label, so that a correct anomaly flag corresponds to recognising previously unobserved usage patterns.

### 4.1. Clustering methods evaluation

We assessed clustering quality with two internal metrics—Silhouette Coefficient (SC) and Davies–Bouldin Index (DB)—plus three external metrics: Adjusted Rand Index (ARI), Adjusted Mutual Information (AMI) and V-Measure [21, 22, 23, 24].

Results in Table 1 show that OPTICS and HDBSCAN lead the pack, each achieving an Adjusted Mutual Information of 0.80, a Silhouette Coefficient of 0.78, a V-Measure of 0.81 and the lowest Davies–Bouldin Index (0.39), indicating particularly well-separated, compact clusters. DBSCAN follows closely, demonstrating very similar clustering quality with slightly less compactness. BIRCH, while still respectable, exhibits a higher Davies–Bouldin Index (0.53), suggesting that its cluster boundaries are marginally less distinct in this application.

*Table 1. Clustering methods evaluation metrics.*

| ***Method*** | ***Hyperparameters*** | ***AMI*** | ***SC*** | ***ARI*** | ***V-Measure*** | ***DB*** |
|---|---|---|---|---|---|---|
| *DBSCAN* | $\varepsilon = 0.7$<br>$m_p = 5$ | *0.79* | *0.77* | *0.69* | *0.80* | *0.45* |
| *OPTICS* | $m_p = 3$<br>$\xi = 0.05$<br>$m_{cs} = 20$<br>$\varepsilon = 0.5$ | *0.80* | *0.78* | *0.69* | *0.81* | *0.39* |
| *HDBSCAN* | $m_{cs} = 5$<br>$m_p = 3$ | *0.80* | *0.78* | *0.68* | *0.81* | *0.39* |
| *BIRCH* | *T = 0.7*<br>*B = 25* | *0.78* | *0.76* | *0.68* | *0.79* | *0.53* |

### 4.2. Comparison methods evaluation

The core samples $\mathcal{C}$ obtained from training dataset using the clustering algorithm with the best evaluation metrics are passed to the comparison engine. The comparison algorithm then compared every data point of test dataset and predicted its anomaly or normal label based on similarities to core samples. Heatmaps were generated with specific states, allowing us to know their labels in advance. And we use these labels to evaluate the comparison method's effectiveness using accuracy, precision, recall and F1-score metrics. Because the anomaly label is tied exclusively to the hidden predefined states, recall is the most direct indicator of successful anomaly detection.

Results of the performed comparison methods tests is shown in Table 2. Both Nearest Neighbour and One-Class SVM achieve identical detection outcomes accuracy of 0.83 with perfect recall and an F1-score of 0.82 indicating strong sensitivity to anomalies albeit with moderate precision. A recall of 1.00 for both Nearest Neighbour and One-Class SVM therefore means that every synthetic malicious minute was correctly flagged by the system. By contrast, Isolation Forest exhibits higher precision (0.89) but markedly lower recall (0.38), reflecting a more conservative detection profile. Consequently, the distance and margin-based methods outperform the tree-based approach under the present evaluation criteria.

*Table 2. Comparison methods evaluation metrics.*

| ***Method*** | ***Hyperparameters*** | ***Accuracy*** | ***Recall*** | ***Precision*** | ***F1 Score*** |
|---|---|---|---|---|---|
| *Nearest Neighbour* | $\varepsilon = 0.4$ | *0.83* | *1.00* | *0.70* | *0.82* |
| *One-class SVM* | $\nu = 0.01$ kernel = linear | *0.83* | *1.00* | *0.70* | *0.82* |
| *Isolation Forest* | $n_{\text{est}} = 25$ cont. $= 0.05$ | *0.75* | *0.38* | *0.89* | *0.53* |

## 5. CONCLUSIONS AND FUTURE WORKS

In this study, we presented a novel two-stage framework for detecting behavioural baseline deviations in endpoint usage by analysing mouse dynamics. In the training phase, non-parametric clustering algorithms were applied to synthetic, state-labelled heatmap data in order to extract core samples that characterise distinct operational modes. We demonstrated that OPTICS and HDBSCAN yield the most compact and well-separated clusters, while BIRCH and DBSCAN performed closely behind. In the detection phase, each new heatmap embedding was compared against the core samples via three paradigms distance thresholding, margin-based classification and ensemble isolation. Both Nearest Neighbour and One-Class SVM achieved perfect recall (1.00) and high F1-scores (0.82), indicating excellent sensitivity to novel patterns, whereas Isolation Forest favoured precision (0.89) at the expense of recall (0.38). In other words, the proposed pipeline not only model's routine behaviour but also pinpoints every previously unseen and intentionally malicious state injected into the evaluation timeline, validating its suitability for real-time security monitoring under adversarial conditions.

Future work will focus on integrating additional contextual features such as keystroke dynamics, application focus. We also plan to explore adaptive, online clustering schemes that update core samples incrementally, and to investigate deep-learning approaches for richer behavioural representations. Finally, automated hyperparameter selection perhaps via Bayesian optimisation and user-specific calibration may further enhance detection robustness and reduce false alarms in deployed endpoint security systems.

## REFERENCES

[1] Al-Asli, M, Ghaleb, T.A. Review of signature-based techniques in antivirus products. *2019 International Conference on Computer and Information Sciences (ICCIS)*, Sakaka, Saudi Arabia, 2019, pp. 1-6. DOI: 10.1109/ICCISci.2019.8716381

[2] Davidian, M., Kiperberg, M., Vanetik, N. Early ransomware detection with deep learning models. *Future Internet*, vol.16, no.8, 2024, 291. DOI: 10.3390/fi16080291

[3] Lee, SJ., Jeon, SE., Lee, IG. A machine learning-enhanced endpoint detection and response framework for fast and proactive defence against advanced cyber-attacks. *Soft Computing*, vol. 28, 2024, pp. 7807-7821. DOI: 10.1007/s00500-024-09727-7

[4] Asgarov, K. Real-time endpoint anomaly detection using adaptive statistical methods for baseline deviations. *Problems of Information Technology*, vol. 16, no. 1, 2025, pp. 11-17. DOI: 10.25045/jpit.v16.i1.02

[5] Martin, A. G., Beltran, M., Fernandez-Isabel A., de Diego I.M. An approach to detect user behaviour anomalies within identity federations. *Computers & Security*, vol. 108, 2021. DOI: 10.1016/j.cose.2021.102356

[6] Chong, P., Elovici, Y., Binder, User Authentication based on mouse dynamics using deep neural networks: A comprehensive study. *IEEE Transactions on Information Forensics and Security*, vol. 15, 2020, pp. 1086-1101. DOI: 10.1109/TIFS.2019.2930429

[7] Shi, Y., Wang, X., Zheng, K. User authentication method based on keystroke dynamics and mouse dynamics using HDA. *Multimedia Systems*, vol. 29, 2023. DOI: 10.1007/s00530-022-00997-5

[8] Matthiesen, J. J. Brefeld, U. Assessing user behaviour by mouse movements. *Stephanidis, C., Antona, M. (eds) HCI International 2020 - Posters. HCII 2020. Communications in Computer and Information Science*, vol. 1224, 2020. DOI: 10.1007/978-3-030-50726-8_9

[9] Kirsh, I. Using mouse movement heatmaps to visualize user attention to words. *Proceedings of the 11th Nordic Conference on Human-Computer Interaction: Shaping Experiences, Shaping Society (NordiCHI '20)*, article 117, 2020, 1-5. DOI: 10.1007/978-3-642-36318-4_3

[10] Kuric, E., Demcak, P., Nemcek, P. Is mouse dynamics information credible for user behaviour research? An empirical investigation. *Computer Standards & Interfaces*, vol. 90, 2024. DOI: 10.1016/j.csi.2024.103849

[11] Khan, S., Devlen, C., Manno, M. Hou, D. Mouse dynamics behavioral biometrics: A survey. *ACM Computing Surveys*, vol. 56, 2024, pp. 1-33. DOI: 10.1145/3640311

[12] Bontempi, G., Taieb, S.B. Borgne, Y-A. Machine learning strategies for time series forecasting. *Business Intelligence: Second European Summer School*, 2013, pp. 62-77. DOI: 10.48550/arXiv.2103.08756

[13] Li, Y., Chen, Y., Dai, X., Liu, M., Chen, D., Yu, Y., Yuan, L., Liu, Z., Chen, M., Vasconcelos, N. Revisiting dynamic convolution via matrix decomposition. *International Conference on Learning Representations*, 2021. DOI: 10.48550/arXiv.2103.08756

[14] Ester, M., Kriegel, H-P., Sander, J., Xu, X.. A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, 1996.

[15] Ankerst, M., Breunig, M. M., Kriegel, H.-P., Sander, J. OPTICS: ordering points to identify the clustering structure. *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, 1999, pp. 49–60. DOI: 10.1145/304182.304187

[16] Campello, R. J. G. B., Moulavi, D., Sander, J., Zimek, A. Hierarchical density estimates for data clustering, visualization, and outlier detection. *Data Mining and Knowledge Discovery*, vol. 10, no. 5, 2015, pp. 1–51. DOI: 10.1145/2733381

[17] Zhang, T., Ramakrishnan, R. Livny, M. BIRCH: an efficient data clustering method for very large databases. *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, 1996, pp. 103–114. DOI: 10.1145/233269.233324

[18] Schoellkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., Williamson, R. C. Estimating the support of a high-dimensional distribution. *Neural Computation,* vol.13, no.7, 2001, pp.1443-1471. DOI: 10.1162/089976601750264965

[19] Liu, F. T., Ting, K. M., Zhou, Z. H. Isolation Forest. *Proceedings of the 2008 IEEE International Conference on Data Mining*, 2008, pp. 413-422, IEEE. DOI: 10.1109/ICDM.2008.17

[20] Asgarov, K. Anomaly mouse dynamics repository. *GitHub, 2025*, URL: https://github.com/tivole/anomaly-mouse-dynamics.

[21] Rousseeuw P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics,* 1987, p. 53-65. DOI: 10.1016/0377-0427(87)90125-7

[22] Davies D. L., Bouldin D.W. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1979, pp. 224-227. DOI: 10.1109/TPAMI.1979.4766909

[23] Vinh, N. X., Epps, J., Bailey, J. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, vol. 11, 2010, pp. 2837-2854.

[24] Rosenberg, A., Hirschberg, J. V-Measure: A conditional entropy-based external cluster evaluation measure. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2007, pp.410-420.

## *Information about the authors:*

**Kamran N. Asgarov** – Graduated with honours from Lomonosov Moscow State University Baku Branch, where he earned both his bachelor's and master's degrees. Currently, he is pursuing a Ph.D. at Azerbaijan Technical University and working as a senior software engineer at R.I.S.K. Company. Research interests are anomaly detection, machine learning, reinforcement learning, and information security.

**Manuscript received on 01 June 2025**