Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 18 No. 3 (2024) | OPEN ACCESS

https://doi.org/10.3991/ijim.v18i03.44467

PAPER

Proposed Model for Real-Time Anomaly Detection in Big IoT Sensor Data for Smart City

Zirije Hasani, Samedin Krrabaj(✉), Marigona Krasniqi

University "Ukshin Hoti," Prizren, Kosovo

samedin.krrabaj@uni-prizren.com

ABSTRACT

A smart city represents an advanced urban environment that utilizes digital technologies to improve the well-being of residents, efficiently manage urban operations, and prioritize long-term sustainability. These technologically advanced cities collect significant data through various Internet of Things (IoT) sensors, highlighting the crucial importance of detecting anomalies to ensure both efficient operation and security. However, real-time identification of anomalies presents challenges due to the sheer volume, rapidity, and diversity of the data streams. This manuscript introduces an innovative framework designed for the immediate detection of anomalies within extensive IoT sensor data in the context of a smart city. Our proposed approach integrates a combination of unsupervised machine learning techniques, statistical analysis, and expert feature engineering to achieve real-time anomaly detection. Through an empirical assessment of a practical dataset obtained from a smart city environment, we demonstrate that our model outperforms established techniques for anomaly detection.

KEYWORDS

smart city, Internet of Things (IoT), real time, anomaly detection, big data

1 INTRODUCTION

Real-time anomaly detection is of paramount significance across diverse domains. Its significance is particularly amplified in the realm of Internet of Things (IoT) data, as real-time anomaly identification has the potential to yield multifaceted benefits for various institutions, enabling them to make informed decisions and anticipate potential challenges. This research focuses on the detailed analysis of IoT sensor data, particularly from sensors within smart city environments. It introduces a groundbreaking model designed for real-time anomaly detection in the data streams generated by these sensors.

Building on our previous study [1], we have introduced a real-time big data anomaly detection algorithm called HW-GA. In this current study, we subjected

Hasani, Z., Krrabaj, S., Krasniqi, M. (2024). Proposed Model for Real-Time Anomaly Detection in Big IoT Sensor Data for Smart City. *International Journal of Interactive Mobile Technologies (iJIM)*, 18(3), pp. 32–44. https://doi.org/10.3991/ijim.v18i03.44467

Article submitted 2023-08-28. Revision uploaded 2023-09-30. Final acceptance 2023-11-24.

the algorithm to rigorous testing by using a dataset that includes over 100,000 data entries from four distinct sensor types.

The main contribution of this paper lies in several key aspects. It entails a comparative analysis of the previously introduced algorithm [1], assessing its effectiveness in real-time versus non-real-time scenarios to identify potential performance differences, considering the urgent requirement for quick processing. Subsequently, our algorithm is compared to alternative iterations, such as HW calc. MASE, HW def. MASE(k), and HW definition. MASE(k, n). Furthermore, we explore the impact of data volume on algorithm performance, explaining how our algorithm strategically breaks down data into windows for continuous analysis instead of processing the entire dataset concurrently, as is typical in batch environments.

In contrast to our previous research [1], this paper innovatively compares the algorithm's real-time and non-real-time executions, supported by detailed execution time calculations.

When dealing with significant data inflows, it is essential to consider three foundational attributes: quantity, accuracy, and diversity. Nonetheless, recent discoveries emphasize the increased importance of speed and relevance. Celerity refers to the speed of data generation and processing, requiring efficient performance evaluation to ensure quick data handling. Furthermore, the inherent value of data in generating insights and making an impact has become a crucial aspect. This has expanded the scope of performance assessment by incorporating methodologies from diverse fields. Optimal algorithm selection depends on a thorough understanding of the analyzed data, including its speed, importance, and the previously mentioned aspects of quantity, accuracy, and diversity. This comprehensive understanding facilitates well-informed decision-making and optimizes data processing for valuable results.

This study utilizes a combination of qualitative and quantitative research methodologies. Qualitative methods are fundamental to a literature review, as they explore existing theories related to the performance evaluation of anomaly detection algorithms in the context of big data. Conversely, quantitative approaches are used to drive experiments aimed at assessing the execution time and categorization efficiency of the HW-GA algorithm.

The research employs a variety of methodological procedures, including analysis, classification, comparison, synthesis, induction, deduction, and experimentation. The first phase involves analyzing the characteristics of large data streams originating from complex computer systems and commercial transactions. Subsequently, existing methodologies and algorithms relevant to anomaly detection within such data streams, especially contextual and collective anomalies, are comprehensively examined. Performance evaluation methodologies include measuring algorithm execution time and conducting comparative tests with both real and simulated data.

Comparative methods are chosen to assess the potential impact of data volume on algorithm performance across different data sizes. Furthermore, a comprehensive analysis of the execution time of the HW-GA algorithm is conducted, with particular focus on its visualization component during the algorithm's culmination.

Inductive and deductive reasoning are used to investigate the potential use of the HW-GA algorithm for real-time anomaly detection in large datasets, especially in massive data streams. The R software environment serves as the platform for statistical analysis, experimentation, and result visualization. Benchmarking

and real-time data from Libelium sensors [2] form the foundation for algorithm validation.

The following sections of this study are organized as follows: Section two delves into relevant literature; Section three introduces standard and real datasets for experimentation; Section four provides a concise overview of the algorithm; Section five navigates through the experimental testing of categorization algorithms; Section six encapsulates findings and conclusions drawn from this study.

2 RELATED WORK

The authors of this paper [1] explain the process of automated anomaly detection. Their approach integrates various concepts, acknowledging the diversity of sensor types and data collection protocols within the context of smart cities.

In the context of anomaly detection, previous research [2] aims to offer a clear explanation of the challenges encountered during data acquisition and anomaly identification. The authors highlight potential issues that may arise from sensor placement and data acquisition, and these issues are systematically addressed through various strategies.

Within this context, another study [3] introduces a domain-agnostic methodology for analyzing real-time data streams from sensor networks. The methodology includes access points to various sensor data sources, each using different protocols, and is customized for smart city applications.

In the specific field of real-time IoT anomaly detection, a specialized architecture has been developed [4]. This architecture integrates five distinct anomaly detection algorithms to identify outliers, followed by a consensus mechanism using a support function to determine if a data point in a time series genuinely qualifies as an anomaly.

In the context of multiple Internet of Things (MIoT) scenarios [5], this paper discusses two central components related to anomaly classification. The first involves a novel methodological framework that is conducive to future explorations in the field, while the second pertains to extending anomaly detection techniques.

An innovative GEER-DLAD technique, introduced in reference [6], enhances IoT applications. This technique involves data collection by IoT devices, followed by compression using the ELC technique. Subsequently, the devices utilize the MSO algorithm for routing, ultimately selecting the best route to the destination.

The research outlined in this paper [7] introduces a labeled Internet of Things (IoT) dataset designed for anomaly detection. This dataset includes initial dynamic adaptive detection (DAD) alongside real-world traffic data, covering various anomaly scenarios and well-defined feature extraction techniques. Machine learning algorithms are then used to detect anomalies from IoT devices.

Exploring the intersection of anomaly detection and smart agriculture [8], this research emphasizes the importance of anomaly detection in this field. The study explores the MLR and LSTM algorithms, which can be synergistically combined to achieve optimal results.

Examining anomaly detection solutions in the IoT [9], this study highlights the prevalent use of hash-based or log-based comparisons against expected behaviors, often overlooking hidden anomalies.

Incorporating IoT advancements into urban settings, a novel approach is introduced [10] that utilizes observation to gather data for proactive weather hazard

identification. This approach helps save human lives and minimize property loss by recognizing hazards early.

Building on existing literature [11, 14], this study explores anomaly detection in decentralized transmission frequency management systems. IoT devices are vulnerable to attacks that manipulate their transmission frequency and data stream tempo. Different manipulation strategies that alter transmission frequency are discussed.

Smart cities are at the forefront of IoT research, and this paper [12, 13] introduces an automated anomaly detection system for IoT solutions. The authors skillfully identify various types of anomalies in diverse contexts, including mobile devices, vehicles, and air sensors, using data collected from a variety of sensors.

This review presents a comprehensive classification of the variety of algorithms for anomaly detection [15, 16]. Three primary methods are highlighted: testing, semi-testing, and obtaining unavailable data. Each category involves specific algorithms, such as support vector machines (SVM) and artificial neural networks (ANN), for fault detection. Semi-supervised error detection is also explored, using techniques such as autoencoders, Gaussian models, and kernel density estimation.

Managerial functions are crucial in the context of smart cities and the IoT fog computing [17–21, 29, 30] has emerged as an efficient approach to meet the stringent requirements of smart cities. Its role includes anomaly detection and ensuring secure data communication during transmission.

3 BENCHMARK DATASETS AND TESTING ENVIRONMENTS

In this section of the paper, we explore the extensive dataset derived from diverse sensors on different days, illuminating the detected fluctuations within this data. It is a universally acknowledged fact that maintaining consistent data replication across multiple days is unfeasible due to the inherent variability in natural elements such as temperature and air quality. The dataset in question relates to the Libelium sensor and includes readings on air quality, soil conditions, temperature, noise levels, and other factors. In Table 1, we have presented a small portion of our data. The Libelium smart-city and smart agriculture sensors have been installed in the city of Prizren by our team to gather data for our research. Data is collected every five minutes, every day starting in 2022, and stored in a data center located at the Innovation and Training Park (ITP) in Prizren. We access and monitor data collection in real time. Now there are over 200,000 rows, and this number is increasing. These readings are presented in a time series format, with each day's data showing a cumulative count of instances. Commencing in March 2022, this dataset comprises two primary attributes: the timestamp of the reading and the corresponding measured value. The dataset specific to Smart City Pro is visually represented through graphs that distinctly highlight instances of real-time anomalies.

We tested the dataset using the HW-GA algorithm to determine its ability to autonomously detect anomalies even in the absence of predetermined anomaly intervals. Our approach follows the logic of the training and test sets, thus simulating real-world conditions. The following illustration depicts the noise data, with anomalies visually marked by encircled red dots.

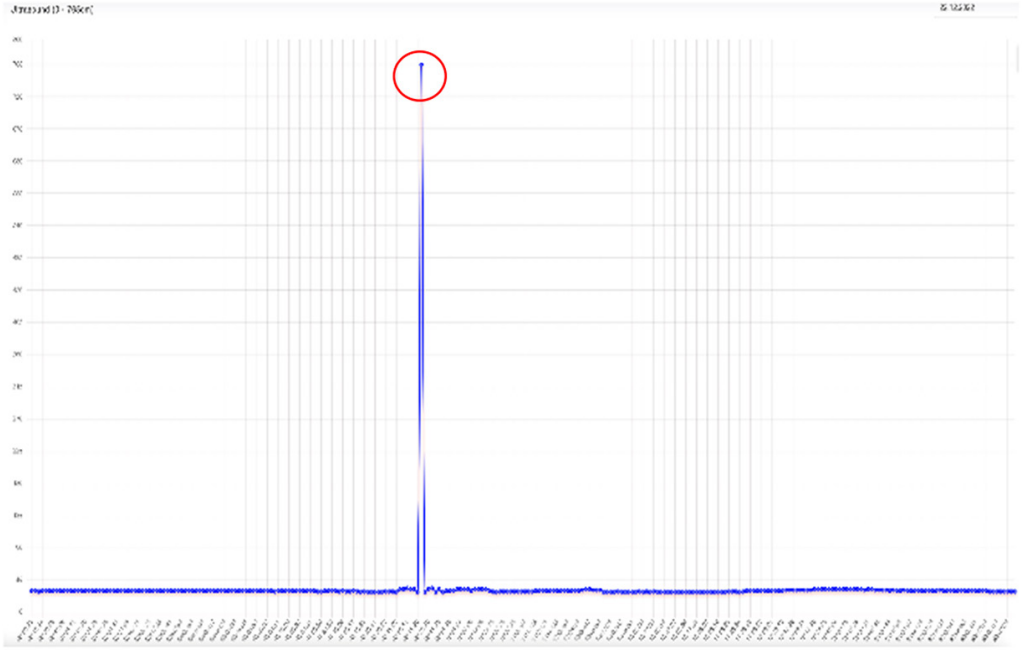

Fig. 1. Anomalies in ultrasound data

The following dataset contains real-time data used for our experimental analysis. Upon inspection, you will observe instances where values remain consistent within a specific timeframe. Conversely, there are cases where values significantly deviate from the preceding ones, as presented in Figure 1, indicating the occurrence of an anomaly and erroneous data. The algorithm used in the experiment demonstrates that anomalies are frequently linked to significant fluctuations in value. Notably, issues arise when sensors are reset, causing values to spike dramatically after the restart. Our goal is to assess the effectiveness of the algorithm in detecting these anomalies, especially in cases where significant shifts in values occur.

Table 1. Part of the benchmark data used for experiments

LUX		US		TC		NOISE	
Timestamp	Value	Timestamp	Value	Timestamp	Value	Timestamp	Value
23/12/2022 12:02:26	192	28/12/2022 00:15:08	765	27/12/2022 12:06:23	15.85	28/12/2022 09:33:26	81.18
23/12/2022 12:23:05	192	28/12/2022 00:25:14	37	27/12/2022 12:31:58	18.01	28/12/2022 09:43:34	51.67
23/12/2022 12:38:26	210	28/12/2022 00:35:20	30	27/12/2022 12:57:30	19.25	28/12/2022 09:53:46	55.06

4 ARCHITECTURE OF THE PROPOSED MODEL FOR REAL-TIME

The proposed framework for real-time anomaly detection within the context of big IoT sensor data for smart cities includes the following sequential stages:

Step 1: Data preprocessing: During this initial phase, the raw data from IoT sensors undergoes preprocessing. This process involves data cleansing, normalization, and feature extraction. A variety of methods, such as smoothing, filtering, and resampling, are used to remove noise from the data. We experience periods when

our sensors encounter problems and fail to collect data due to various reasons, such as Internet connection issues, power outages, etc. In that case, the missing data is replaced with random data that is closer to the expected value for that time period. The data is extracted in .csv format for analysis.

Step 2: Feature engineering: In the next step, relevant features are extracted from the preprocessed data. This extraction combines domain-specific knowledge with statistical analysis to identify the most relevant attributes for effective anomaly detection.

Step 3: Anomaly detection: This stage involves using the HW-GA anomaly detection algorithm [27] and its modified versions to identify anomalies in real time.

The architecture, depicted in Figure 2, outlines the process from data reception through sensors to its presentation on the user or client interface. The process is divided into distinct phases: data acquisition from sensors (specifically Waspmote devices), followed by communication through the XBEE 802.15.4 protocol within Meshlium. A notable feature of this architecture is the external database that stores the data, which is also synchronized with a local database. The description so far has covered the flow from receiving data to storing and synchronizing it.

A web application has been developed to visualize this data through graphs, aiming to improve user comprehension. This application is built using the PHP programming language, with CodeIgniter4 as its underlying framework, following the Model-View-Controller paradigm. The client initiates a request for specific data in the view layer using a graph. Subsequently, the view interacts with the controller to convey the database query (model). The process reverses as the controller relays the model's response to the view and presents the requested data to the client. After receiving the client's request, the data is subjected to anomaly testing using the HW-GA algorithm, and the results are visually depicted.

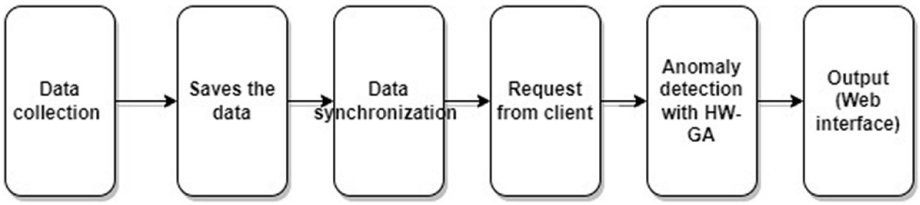

Fig. 2. Architecture of proposed model for real-time anomaly detection

4.1 Data collection

The data collection process is initiated by a device called Waspmote, which directly interfaces with sensors to retrieve data. Waspmote represents an original hardware platform, providing developers with complete control over its functionalities [22]. This platform provides physical access to the board, allowing for the addition of new sensors or integration into various electronic sensor devices. Waspmote consists of distinct components: some are dedicated to node control, others facilitate sensor connectivity, and certain parts serve as identification elements. Each Waspmote model is associated with stickers indicating its application domain, such as smart cities, smart parking, smart environment, and others [22].

Communication between Waspmote and Meshlium is achieved through the XBee 802.15.4 protocol [23]. Meshlium serves as an IoT gateway, capable of supporting up to four different radio interfaces. These interfaces include a 2.4 GHz WiFi access point, a 4G/3G/GPRS/GSM module, and two XBee/RF radios [24].

4.2 Data synchronization

Meshlium stores data in a local database and then synchronizes the table in this local database with an external database. To access the Meshlium database from an external application, specific parameters need to be used within the designated section [25]. Various MySQL management applications, such as MySQL Workbench or SQLyog, can be used to interact with the database on Meshlium and perform maintenance tasks. Additionally, Meshlium includes a pre-installed instance of phpMyAdmin, which facilitates the management of the local database [25].

4.3 Request from the client

Client requests are processed through the primary phases of the controller and model. The controller, serving as an intermediary between the model and view components, undertakes crucial tasks such as processing input requests (often generated by client clicks), managing data using the model, and collaborating with the view to showcase the final outcomes [26]. Conversely, the model component is responsible for the underlying logic that governs the data with which the user or client interacts [26].

4.4 Output (web interface)

Involving the final stage of the web application, this section pertains to the client's action of clicking, which effectively initiates a request within the database. This request is then displayed on the interface. The client view component, also known as UI logic, encompasses all the elements of the user interface that directly interact with the end user [26].

5 ALGORITHMS USED FOR TESTING

Citing previous research [11], the authors investigated anomaly detection in diverse scenarios, utilizing both a mathematical rule-based approach and an LSTM-based approach, particularly within a decentralized transmission frequency management system. They explained the vulnerabilities of IoT devices to attacks that manipulate their transmission frequency, resulting in an incorrect data stream tempo. As various manipulations can alter transmission frequencies, this becomes a significant concern.

Subsequently, a comparative analysis was conducted involving our algorithm and alternative iterations such as HW calc. MASE, a measure of the accuracy of a forecast MASE(k), and HW definitions MASE(k, n) [27]. The objective was to assess whether our proposed method demonstrates superior performance compared to alternative versions of the algorithm, especially in a real-time environment.

6 PERFORMANCE AND PRECISION MEASUREMENT OF HW-GA ANOMALY DETECTION ALGORITHM IN REAL TIME

The algorithm examined this study is the HW-GA algorithm [27]. The objective of this investigation is to compare real-time data with non-real-time data (static time).

The algorithm was tested in two distinct ways, and the resulting Table 2 provides a comparative analysis of the results, measured in seconds.

Table 2. Experiments with real-time and non-real-time data

Algorithms	LUX		US		TC		NOISE	
	Execution Time (Seconds)		Execution Time (Seconds)		Execution Time (Seconds)		Execution Time (Seconds)	
	Real-Time	Static Time	Real Time	Static Time	Real Time	Static Time	Real Time	Static Time
HW GA	1.023	3.124	1.001	3.451	0.124	1.011	0.289	2.111
HW calc. MASE	1.145	3.315	1.024	4.156	0.213	1.815	0.426	2.312
HW def. MASE(k)	1.213	3.512	1.146	3.012	0.315	2.415	0.515	2.506
HW def. MASE(k, n)	1.192	3.301	1.289	3.412	0.918	2.956	0.819	2.415

As shown in Table 2, the real-time data demonstrates faster execution during testing compared to the non-real-time data. This disparity in results is also visually represented by the accompanying graphs. The evaluation of the algorithm focuses on the processing of this data. To accomplish this, the algorithms are implemented using the R programming language, seamlessly integrated with CodeIgniter.

The code includes a time measurement mechanism at the beginning and end. This procedure provides the execution time of the algorithm, providing valuable insights into its efficiency.

What becomes apparent is that the experiment shows that real-time data has a higher speed compared to non-real-time data.

The graph in Figure 3 illustrates a comparison of the execution times of various algorithms: HW-GA [27] algorithm, HW-calculated MASE [27], HW-GA-defined MASE(k) [27], and HW-GA-defined MASE(k, n) [27], in both real-time and non-real-time anomaly detection scenarios. The graph effectively illustrates that real-time anomaly detection is faster than non-real-time detection.

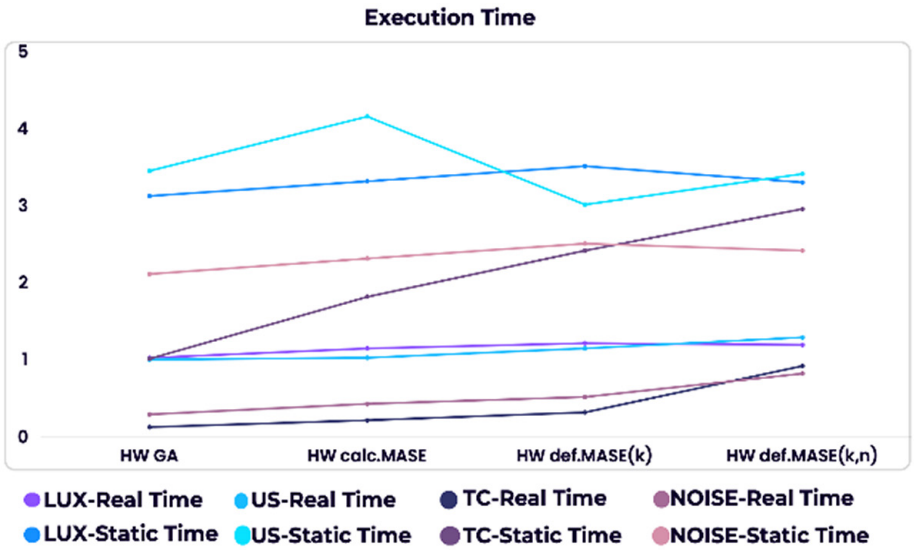

Fig. 3. Experimentation comparing real data time and non-real data time in seconds

7 RESULT DISCUSSION

In this section, we present the evaluation results of our proposed model using a real-world dataset obtained from a smart city environment. This dataset contains data collected from different sensors, such as temperature, noise, and humidity. Our main goal is to evaluate the accuracy and effectiveness of our previously introduced algorithm, HW-GA [27], in comparison to various variations of the same algorithm, such as HW-calculated MASE [27], HW-GA-defined MASE(k) [27], and HW-GA-defined MASE(k, n) [27]. Specifically, we evaluate the effectiveness of our proposed algorithm in identifying true positive (TP) anomalies in both real-time and non-real-time scenarios. We particularly focus on understanding the potential impact of real-time environmental conditions on the algorithm's execution time.

During our experiments with four datasets, we faced challenges in identifying anomalies. The experiments with static data took between 1 and 3 seconds to complete, while real-time experiments showed significantly faster anomaly detection capabilities. To conduct these evaluations, we implemented the HW-GA [27] algorithm, HW-calculated MASE [27], HW-GA-defined MASE(k) [27], and HW-GA-defined MASE(k, n) [27] using the R programming language integrated within the CodeIgniter framework.

The results reveal that the execution time of the HW-GA algorithm varied significantly depending on the dataset and the presence of real-time data. For example, the HW-GA algorithm processed the LUX dataset in just 1.023 seconds in real-time conditions, whereas the same dataset required 3.124 seconds under static conditions. This significant difference in execution speed highlights the impact of data volume and the number of anomalies to be detected on execution time. In the context of extensive datasets, effective visualization becomes crucial for identifying numerous anomalies.

Our analysis highlights the need to enhance the HW-GA algorithm by incorporating additional parameters into the anomaly detection process. This refinement aims to evaluate the impact of these parameters on execution time. The fundamental principle of our proposed algorithm is to identify the optimal parameter values that guarantee the accurate computation of true positives (TP), false positives (FP), and false negatives (FN). The assessment of these parameters depends on the characteristics of the dataset and the duration over which anomalies are being sought.

To evaluate the precision of our algorithm, we employ the following metrics:

- TP (true positive): Instances where anomalies are correctly detected within annotated intervals
- FP (false positive): Instances where anomalies are mistakenly identified outside annotated intervals
- FN (false negative): Cases where annotated intervals contain undetected anomalies

For this assessment, our genetic algorithm optimization uses an evaluation function that was detailed in previous research [28]. To facilitate result comparison, we rely on two significant statistical metrics commonly used in binary classification testing: the detection rate (recall), expressed as a percentage (d.r.), and precision (prec.). These metrics provide valuable insights into the performance of our algorithm. Due to the significant number of TN values, metrics such as specificity and accuracy are not suitable for evaluating time series data. The detailed results are presented in Table 3.

Table 3. The result from all tested algorithms

	LUX					US					TC					NOISE				
	TP	TN	FP	d.r.	prec.	TP	TN	FP	d.r.	prec.	TP	TN	FP	d.r.	prec.	TP	TN	FP	d.r.	prec.
HW calc. MASE	1	0	0	100	100	0	1	0	100	100	1	0	33	–	0	0	1	0	100	100
HW def. MASE(k)	1	0	0	100	100	0	1	0	100	100	1	0	34	–	0	0	1	0	100	100
HW def. MASE(k,n)	0	0	0	100	100	0	1	3	100	28	1	0	1	–	0	0	1	0	100	100
HW-GA	1	1	0	100	100	0	1	0	100	100	1	0	0	–	–	0	1	0	100	100

8 CONCLUSION

In this research study, we present an innovative model specifically designed for real-time anomaly detection in the context of big IoT sensor data for smart cities. The proposed model combines unsupervised machine learning techniques, statistical analysis, and feature engineering to effectively detect anomalies in real-time settings. We proceeded to evaluate the performance of our model using a real-world dataset obtained from a smart city environment. The results of our evaluation demonstrate that our model exceeds the capabilities of existing anomaly detection methods. By effectively detecting anomalies in real time, our proposed model contributes to enhancing the operational efficiency and security of smart cities.

Assessing the effectiveness of real-time anomaly detection algorithms in the context of big data is crucial, as timely processing is essential for managing real-time data. The study has two main objectives: firstly, to examine the impact of the HW-GA algorithm on various data types; and secondly, to assess the algorithm's performance when tested on datasets with more than 100,000 rows to determine its scalability. Our findings indicate that the HW-GA algorithm demonstrates efficiency in terms of execution time, especially in real-time scenarios where it quickly identifies anomalies. The algorithm performs exceptionally well in specific performance parameters, such as execution speed and accuracy in anomaly detection, which also influence the volume of data within the realm of big data.

Across various testing environments, we consistently observe that the algorithm operates more efficiently with real-time data. These outcomes are presented in the tables provided (Tables 2 and 3). Furthermore, we conclude that visualizing the results of the HW-GA algorithm is essential only when anomalies appear within the initial 50% of the testing data. However, in situations where anomalies are not present within this percentage, visualization becomes unnecessary.

In the near future, we intend to refine and update the algorithm to conduct thorough and accurate tests promptly. This endeavor aims to achieve optimal results, considering the importance of anomaly detection in the advancement of large-scale data analysis.

9 FUTURE WORKS

Various methods exist for detecting anomalies. In this paper, we have explained a technique that utilizes an algorithm developed in the R programming language and integrates it within a web application using the CodeIgniter framework. While sensors continue to collect and transmit real-time data, the critical aspect lies

in testing this data to identify anomalies in real time. The identification of anomalies serves as an indicator of data collection issues, and once a specific anomaly is recognized, corrective measures should be implemented.

This paper has focused on extensively testing a single algorithm and its associated categorizations. However, looking ahead, the approach will expand to encompass broader horizons. The incoming real-time data will be tested using a variety of algorithms developed in different programming languages to improve the range and accuracy of anomaly detection.

10 REFERENCES

[1] Pierfrancesco Bellini, Daniele Cenni, Paolo Nesi, and Mirco Soderi, "Anomaly detection on IOT data for smart city," in *IEEE International Conference on Smart Computing (SMARTCOMP)*, pp. 416–421, 2020. https://doi.org/10.1109/SMARTCOMP50058.2020.00087

[2] Redhwan Al-amri, Raja Kumar Murugesan, Mustafa Man, Alaa Fareed Abdulateef, Mohammed A. Al-Sharafi, and Ammar Ahmed Alkahtani, "A review of machine learning and deep learning techniques for anomaly detection in IoT data," *Applied Sciences*, vol. 11, no. 12, 2021. [Online]. Available: https://doi.org/10.3390/app11125320. [Accessed 20. 12. 2022].

[3] Sergio Trilles, Òscar Belmonte, Sven Schade, and Joaquìn Huerta, "A domain-independent methodology to analyze IoT data streams in real-time. A proof of concept implementation for anomaly detection from environmental data," *International Journal of Digital Earth*, vol. 10, no. 1, pp. 103–120, 2017. [Online]. Available: https://doi.org/10.1080/17538947.2016.1209583. [Accessed 20. 12. 2022].

[4] Elena-Simona Apostol, Ciprian-Octavian Truică, Florin Pop, and Christian Esposito, "Change point enhanced anomaly detection for IoT time series data," *Water*, vol. 13, no. 12, 2021. [Online]. Available: https://doi.org/10.3390/w13121633. [Accessed 20. 12. 2022].

[5] Francesco Cauteruccio, Luca Cinelli, Enrico Corradini, Giorgio Terracina, Domenico Ursino, Luca Virgili, Claudio Savaglio, Antonio Liotta, and Giancarlo Fortino, "A framework for anomaly detection and classification in multiple IoT scenarios," *Future Generation Computer Systems*, vol. 114, pp. 322–335, 2021. [Online]. Available: https://doi.org/10.1016/j.future.2020.08.010. [Accessed 22. 12. 2022].

[6] E. Laxmi Lydia, A. Arokiaraj Jovith, A. Francis Saviour Devaraj, Changho Seo, and Gyanendra Prasad Joshi, "Green energy efficient routing with deep learning based anomaly detection for Internet of Things (IoT) communications," *Mathematics*, vol. 9, no. 5, p. 500, 2021. [Online]. Available: https://doi.org/10.3390/math9050500. [Accessed 28. 12. 2022].

[7] Laura Vigoya, Diego Fernandez, Victor Carneiro, and Fidel Cacheda, "Annotated dataset for anomaly detection in a data center with IoT sensors," *Sensors* (Basel, Switzerland), vol. 20, no. 13, 2020. [Online]. Available: https://doi.org/10.3390/s20133745. [Accessed 26. 12. 2022].

[8] C. Catalano, L. Paiano, F. Calabrese, M. Cataldo, L. Mancarella, and F. Tommasi, "Anomaly detection in smart agriculture systems," *Computers in Industry*, vol. 143, 2022. [Online]. Available: https://doi.org/10.1016/j.compind.2022.103750. [Accessed 03. 01. 2023].

[9] Inês Martins, João S. Resende, Patrícia R. Sousa, Simão Silva, Luís Antunes, and João Gama, "Host-based IDS: A review and open issues of an anomaly detection system in IoT," *Future Generation Computer Systems*, vol. 133, pp. 95–113, 2022. [Online]. Available: https://doi.org/10.1016/j.future.2022.03.001. [Accessed 03. 01. 2023].

[10] Sreenivasulu Bolla, R. Anandan, and Subash Thanappan, "Weather forecasting method from sensor transmitted data for smart cities using IoT," *Scientific Programming*, vol. 2022, 2022. [Online]. Available: https://doi.org/10.1155/2022/1426575. [Accessed 09. 01. 2023].

[11] Hongde Wu, Noel E. O'Connor, Jennifer Bruton, Amy Hall, and Mingming Liu, "Real-time anomaly detection for an ADMM-based optimal transmission frequency management system for IoT devices," 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/16/5945. [Accessed 03. 01. 2023].

[12] Pierfrancesco Bellini, "Anomaly detection on IoT data for smart city," [Online]. Available: https://www.snap4city.org/download/video/AnomalyDetection2020.pdf. [Accessed 27. 12. 2022].

[13] "Difference between contextual anomaly and collective anomaly," 2018. [Online]. Available: https://stats.stackexchange.com/questions/323553/difference-between-contextual-anomaly-and-collective-anomaly. [Accessed 29. 12. 2022].

[14] "Service analytics," 2018. [Online]. Available: https://www.repetico.com/card-67787341. [Accessed 27. 12. 2022].

[15] S. Garg, "Algorithm selection for anomaly detection," 2020. [Online]. Available: https://medium.com/analytics-vidhya/algorithm-selection-for-anomaly-detection-ef193fd0d6d1. [Accessed 27. 12. 2022].

[16] Ayan Chatterjee and Bestoun S. Ahmed, "IoT anomaly detection methods and applications: A survey," 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542660522000622. [Accessed 04. 01. 2023].

[17] B. R. Roy, "OneClassSVM, A density based outlier / Anomaly detection," 2021. [Online]. Available: https://towardsdev.com/oneclasssvm-cfffbc462d2d. [Accessed 26. 12. 2022].

[18] Roy Jafari, Mojtaba Khanzadeh, Brian K. Smith, and Linkan Bian, "Self-organizing and error driven (SOED) artificial neural network for smarter classifications," 2017. [Online]. Available: https://www.researchgate.net/publication/316266796_Self-Organizing_and_Error_Driven_SOED_Artificial_Neural_Network_for_Smarter_Classifications. [Accessed 16. 12. 2022].

[19] "Building autoencoders in keras," 2016. [Online]. Available: https://blog.keras.io/building-autoencoders-in-keras.html. [Accessed 25. 12. 2022].

[20] Z. Li, "Gaussian mixture model: Simple definition," 2011. [Online]. Available: https://www.statisticshowto.com/gaussian-mixture-model/. [Accessed 28. 12. 2022].

[21] José Santos, Philip Leroux, Tim Wauters, Bruno Volckaert, and Filip De Turck, "Anomaly detection for smart city applications over 5G low power wide area networks," in *NOMS 2018 – 2018 IEEE/IFIP Network Operations and Management Symposium*, Taipei, Taiwan, 2018, pp. 1–9. [Online]. Available: https://ieeexplore.ieee.org/document/8406257. [Accessed 09. 01. 2023].

[22] "Waspmote plug and sense technical guide," [Online]. Available: https://www.libelium.com/wp-content/uploads/2013/02/waspmote_plug_and_sense-technical_guide_eng.pdf. [Accessed 17. 01. 2023].

[23] "Xbee 802.15.4 tech brief WhitePaper," [Online]. Available: https://hub.digi.com/dp/path=/marketing/asset/digi-xbee-802-15-4-protocol-comparison-tb#:~:text=the%20FCC%20regulations.-,The%20802.15.,consumed%20with%20the%20occupied%20bandwidth. [Accessed 17. 01. 2023].

[24] "Meshlium technical guide, understanding meshlium," [Online]. Available: https://development.libelium.com/meshlium-technical-guide/understanding. [Accessed 26. 01. 2023].

[25] "Database management," [Online]. Available: https://development.libelium.com/meshlium-technical-guide/dbmanagement. [Accessed 18. 01. 2023].

[26] "MVC Framework – Introduction," [Online]. Available: https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm. [Accessed 16. 01. 2023].
[27] Zirije Hasani, Boro Jakimovski, Goran Velinov, and Margita Konpopovska, "An adaptive anomaly detection algorithm for periodic real time data streams," in *International Conference on Intelligent Data Engineering and Automated Learning (IDEAL)*, 2018. https://doi.org/10.1007/978-3-030-03493-1_41
[28] Tomasz Andrysiak, "Time series forecasting using holt-winters model applied to anomaly detection in network traffic," in *International Joint Conference SOCO17-CISIS17-ICEUTE17*, AISC, 2017, vol. 649, pp. 567–576. https://doi.org/10.1007/978-3-319-67180-2_55
[29] Elizabeth Riddle-Workman, Marina Evangelou, and Niall M. Adams, "Adaptive anomaly detection on network data," [Online]. Available: https://spiral.imperial.ac.uk/bitstream/10044/1/64846/2/AdaptiveAnomalyDetection.pdf. [Accessed 28. 12. 2022].
[30] H. V. Ravinder, "Determining the optimal values of exponential smoothing constants does solver really work?" *American Journal of Business Education*, vol. 6, no. 3, pp. 347–360, 2013. https://doi.org/10.19030/ajbe.v6i3.7815

11 AUTHORS

Zirije Hasani, University "Ukshin Hoti," Prizren, Kosovo (E-mail: zirije.hasani@uni-prizren.com; ORCID: 0000-0001-6888-9465).

Samedin Krrabaj, University "Ukshin Hoti," Prizren, Kosovo (E-mail: samedin.krrabaj@uni-prizren.com).

Marigona Krasniqi, University "Ukshin Hoti," Prizren, Kosovo (E-mail: marigona.krasniqi@uni-prizren.com).