Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 10s (2025)

De-Duplication of Big Data using Immutable Hashing Algorithms for Wearable Heat-Stroke Detection System

Mohamad Emad Bitar[1], Dr. V. Sujatha[2]

[1]Ph.D. Scholar, Department of Computer Science,BharathiarUniversity, CMS College of Science &Commerce Coimbatore, India -*641035t22h.12345@gmail.com*

[2]Vice Principal, CMS College of Science and Commerce,Coimbatore, India –*641035-sujatha.padmakumar4@gmail.com*

Article History:

Received: 12-01-2025

Revised: 15-02-2025

Accepted: 01-03-2025

Abstract:

As global warming progresses, the frequency and intensity of heat waves increase. Heat cramps, heat exhaustion, and heat stroke are just a few of the numerous problems caused by high temperatures. High temperatures frequently cause severe heat stroke without a breeze. Numerous studies have shown that age influences the risk of heat stroke at high temperatures. A wearable device with several biosensors might be used to monitor physiological changes. Although wearable devices are a simple and practical approach to collecting physiological data and providing feedback on the user's body and health statistics, there is no specific method for alerting users to situations that might lead to heat stroke. The temperature values are duplicated in the device after multiple times of heat detection. Continuously monitoring the temperature in the human body, the exponential growth of data is continuous, and the values are stored frequently. So, the storage is filled, and the repeated values are stored as duplicate values in the cloud storage. Therefore, we want to de-duplicate the values to optimize storage resources, enhance data quality, and improve processing efficiency. In this paper, we processed the de-duplication of big data using the proposed Immutable Hashing Algorithm. Where an immutable hashing algorithm is used, we use the Secure Hashing Algorithm (SHA1 with SHA1) method for a better de-duplication process. These methods quickly and securely de-duplicate the data from the Kaggle dataset. The result shows the high reliability and low latency values of de-duplication in the heat stroke prediction process.

Keywords: Big data, De-duplicate, Immutable Hashing Algorithm, Heat Stroke, Temperature monitoring.

I INTRODUCTION

Global warming is one of the most pressing problems caused by heatstroke. This project aims to employ a combination of healthcare Internet of Things (IoT) and machine learning (ML) technologies to create a system that protects the lives of humans and other living things [1-3]. Data de-duplication is a set of strategies used to reduce the amount of data in storage systems by identifying and removing duplicate data [4-6]. Widely used for data backup to minimize bandwidth and storage overhead [7-10]. Depending on the amount of de-duplication, many de-duplication strategies have been proposed, such as file-level or block-level de-duplication. Because of its broad

range of applications, IoT integration with many systems is critical, particularly as technical assistance in areas such as health, which often need real-time information [11-15].

Data de-duplication technology allows cloud clients to effectively manage their cloud storage capacity by eliminating the need to store regular data and saving bandwidth. Finally, the data is stored on the cloud server [16]. The Advanced Encryption Standard (AES) technology saves data in encrypted form, ensuring data privacy [17]. Data de-duplication allows data owners to share a copy of the same data, which reduces storage space utilization [18-22]. The abovementioned challenges motivate research into encrypted data de-duplication [23-26]. In this paper, we provide an encrypted data de-duplication approach that allows for the deletion of duplicate cipher texts, hence improving privacy protection on the cloud storage server.

There are two sorts of data duplication: file and block. Block-level duplication occurs when data blocks spanning a volume are aggregated. Typically, file duplication is considered coarse, whereas block duplication is fine-grained. Block-level de-duplication often yields more results than file-level de-duplication. Block-level block de-duplication works on both identical and comparable files. In our investigation, we developed block-level de-duplication. Block encryption facilitates coordinated de-duplication of standard components. Convergent encryption generates the key value by fragmenting plain text using a hash technique. This paper proposes a detection system that alerts medical practitioners to temperature and heart rate circumstances that may be triggering a heat stroke. This paper proposes efficient de-duplication strategies based on advanced memory management algorithms to overcome issues with large data volumes. Immutable Hashing Algorithms are used to de-duplicate large amounts of data on wearable heat stroke monitoring devices.

II BACKGROUND STUDY

Majumder S. et al. (2017), these authors introduce a detection system that alerts medical personnel to potential heat stroke risk factors, including elevated body temperature and irregular heart rate. The reliability and efficiency of the prototype in transferring humidity and temperature data across the site connection have been confirmed. These writers have made great strides in heat stroke diagnosis, which is crucial in regions where the temperature is high and where people have a good thermal sense of heat so that people may be quickly treated when they are in danger of heat stroke.

Antonio, P. O. (2017). These researchers looked at heat stroke warnings that may be sent to a user's smartphone based on external ambient temperature and humidity data. They determined that age influences the risk of heat stroke in hot weather. A wearable device equipped with several biosensors may be used to monitor physiological changes. Although wearable devices provide a simple and practical method for collecting physiological data and giving feedback about the user's body and health statistics, there is currently no system to accurately warn users to avoid situations that might lead to heat stroke.

Chen, S. T. (2017) these authors examined the load on the organization's processing resources and the cloud provider's bandwidth; a mediator employs data de-duplication on the client side. These authors proposed a new method integrating data de-duplication with cloud data integrity checks using the blockchain. This approach guarantees that both public and private sectors may be audited.

El Ghazouani, M. et al. (2020) these authors examined the de-duplication system integrated the idea of "Proof of Ownership(PoW)" to make it more efficient, but the POW challenges—a series of mathematical tasks that must be solved to confirm ownership—can rapidly exhaust server resources, depending on the file type and size. Significantly cannot change the contents of a file saved using this technique. Therefore, it is only suitable for archiving, even if it sounds efficient.

Hussain, I. (2022) discussed to control deadly cases of heat stroke; these writers devised a wristband system. The wristband has sensors that continuously monitor vital signs, including heat index, oxygen level, and body temperature, and it can also transmit warning signals to wirelessly connected devices in the event of an emergency. Anyone who enjoys being outside and doing a range of activities will find this appropriate. The integrated hardware package with wristbands is also employed for first aid reasons, especially for old and immobile patients.

Rahardja, U. et al. (2021), a hash model of immutability, was suggested by these writers. When keeping track of transactions, many distributed systems turn to blockchain technology. There is no need for intermediaries in blockchain transactions since the databases are widely distributed. To further aid compression, a hash function may reduce the length of an input string by producing a shorter output string. Complete blockchain transactions are immutable due to the reliance on hash algorithms in blockchain systems. The hash links each block in the blockchain to the one that came before it. Therefore, the blockchain is secure and cannot be hacked. To provide a more comprehensive argument, the author suggests expanding the scope of the research beyond the problem of storing immutability on the blockchain.

Table 1: Comparison table of Heat stroke detection using Block-chain

Author & Year	**Focus**	**Method/Technology**	**Application/Use Case**	**Limitations/Remarks**
Majumder, S., et al. (2017)	Heat stroke detection	Temperature and heart rate monitoring	Alert health personnel about potential heat stroke	Efficient and reliable for data transmission, but no feedback system
Antonio, P. O. (2017)	Heat stroke prevention	Wearable devices with biosensors	Sends warnings to users via smartphones based on temperature-humidity data	There is no specific cure for warning customers to avoid dangerous circumstances.
Chen, S. T. (2017)	Data integrity in cloud storage	Block-chain-based de-duplication	Reduces data size and computation time	Ensures data confidentiality but does not focus on heat stroke

El Ghazouani, M., et al. (2020)	Proof of Ownership in de-duplication	Block-chain-based proof of ownership	Data consistency and security in storage systems	Suitable only for archiving; no content updates possible
Hussain, I. (2022)	Heat stroke prevention and care	Wristband with sensors	Measures body temperature, oxygen level, and heat index; sends emergency alerts.	Focuses on bedridden patients and elderly; hardware limitations.
Rahardja, U., et al. (2021)	Blockchain immutability	Hash functions for data integrity	Secure transaction records in decentralized databases	Limited to data storage and integrity; no real-time applications

Son T. W. et al. (2021) these authors proposed a heat stroke detection circuit that uses a fuzzy controller to provide a measurable risk level for heat stroke by monitoring heart rate, ambient temperature, relative humidity, and core body temperature. A wearable IoT-based heat stroke detection device consistently monitors physiological data, whether running or motionless. This gadget notified the user about imminent heat strokes. The Thermal Heat Stroke Risk Coefficient (THSRC) is just as essential as the patient's core temperature and heart rate in determining the risk of heat stroke.

Vyas, A. et al. (2021), these writers provided answers concerning sensitive medical information and other security and privacy issues. To safeguard private medical information, these authors have provided a few instances of methods, algorithms, and techniques that are either already in use or have been developed by researchers in this field. Novel approaches, including autonomous security and privacy protection for healthcare data, are being created in response to the proliferation of AI, ML, and other related technologies. The assessment and preservation of the enormous and diverse medical and healthcare data collection uses many computer technologies and storage systems.

Wijaya N.H. (2020) these authors created a wearable health monitoring system based on the cloud and the Internet of Things (IoT). Wearable IOt-cloud-based health monitoring system (WISE) addresses real-time health monitoring primarily via the BASN (Body Area Sensor Network) viewpoint. Blood pressure, temperature, and pulse sensors are wearable medical equipment available. This will impact the daily usage of smartphones since most existing wearable health monitoring devices depend on them for data processing, visualization, and transmission.

Yin, L. K. et al. (2024), using the Internet of Things (IoT) and machine learning (ML), early detection and prevention of heat-related disorders might be significantly improved. Sensitive Internet

of Things devices may monitor physiological signs like pulse rate, perspiration, and temperature while recording environmental factors like humidity and temperature. Using ML approaches, evaluate and uncover patterns and correlations in the data that show how these characteristics correspond with heat-related illnesses. Predictive algorithms based on this information may alert those in danger in time to seek shade, a drink, or a superb location to sit. Simultaneously, there may be fewer hospital admissions and cases of heat-related sickness. The current study investigates the potential of machine learning and Internet of Things (IoT) devices for predicting heat stroke.

2.1 Problem Identification

Predicting temperature for heat strokes using existing algorithms is performed slowly, and accuracy values are sometimes incorrectly shown. Those algorithms take colossal memory space to store the data and repeatedly store duplicate data. These algorithms are not practical in de-duplication.

III METHODS AND MATERIALS

This paper uses immutable shing algorithms to de-duplicate heat stroke detection data using a Kaggle dataset. Following data collection, the dataset will be preprocessed to prepare it for analysis. A Secure Hash Algorithm (SHA) then processes the essential data to produce a fixed-size hash value for a string of numbers and characters. This step ensures data integrity and makes the de-duplicated dataset.

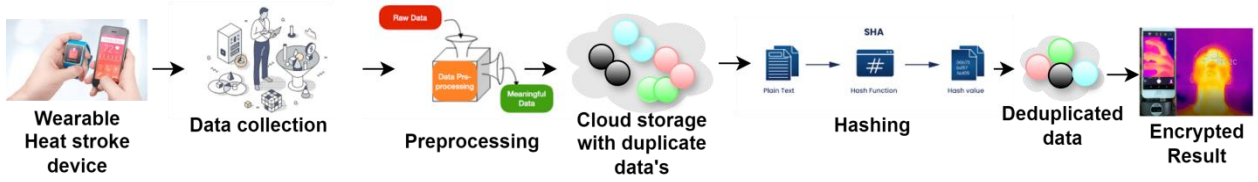

Figure 1: Heat stroke detection Process

3.1 DATASET GATHERING

We have taken the heat stroke data from the Kaggle dataset to evaluate the heat stroke rate and predict the temperature in wearable heat stroke detection system using Immutable Hashing Algorithms. We have taken the Kaggle CSV file for detecting the wearable heat stroke detection system.

Dataset 1: https://www.kaggle.com/datasets/tahiatazin1510997643/heat-stroke

3.2 DATA PREPROCESSING

Data preparation includes integrating, cleaning, and modifying data. Finding relevant information and missing data is the first and most crucial step. Before proceeding with further tasks, data preparation seeks to enhance data quality. To train machine learning models, raw data must first be "prepared" by organizing and transforming it into a suitable format. The goal of this research was to develop a heat stroke dataset that might be used by wearable technologies to identify heat strokes.

3.3 DATA DE-DUPLICATION USING IMMUTABLE HASHING ALGORITHM

Data de-duplication primarily helps to reduce space-consuming and duplicate data. As a result, the data must be analyzed using this method to detect duplicates and remove them from heat stroke databases. Immutable Hashing ensures the secrecy and privacy of health data needed to build models and make predictions in cloud-based machine learning systems for heat stroke detection.

The de-duplication using the Immutable hashing technique detects duplicated data stored in the cloud. Encrypted file storage helps to keep data secure and private in this process. This technique will result in the storing of only files with original content. We built this system using Immutable Hashing techniques as well as encryption approaches. This paper attempts to save users time and space in cloud storage. We de-duplicate quickly and securely using the Secure Hashing Algorithm in (SHA1 with SHA2) Immutable hashing approach. Data de-duplication reduces storage space while reducing the transmission of redundant data divided into blocks; for each block, a hash value is generated. This is generated using the SHA1 with SHA2 method. SHA2 is a 256-bit digest cryptographic hash technique. Attacks using length extension are widespread in SHA1. SHA1, designed as a cryptographic hash algorithm, generates a 160-bit (20-byte) hash output from a given input. It is typically represented as a hexadecimal integer with 40 digits.

Computationally, SHA-1 is faster than SHA-2. SHA-1 was selected for applications that prioritize speed above security in de-duplication processes. Given the established standards' inertia, using SHA1 was simple. Although the extensive Heat Stroke Dataset focuses on speedy duplicate detection rather than high-security requirements, SHA1 may be adequate for de-duplication tasks. SHA1 works effectively in environments with limited resources since its bit length (160-bit hash) requires less memory and processing power. Sometimes, SHA1 files store duplicate files, so we use SHA2 for better results. Many cloud and software systems now configure their hash algorithms to SHA2 for de-duplication. From local file storage systems to cloud platforms, SHA2 is widely supported by all contemporary systems.

Moving to SHA2 ensures security and long-term support and de-duplicates the SHA1 files. SHA2 is expected to be secure in the foreseeable future. This makes it a reliable choice for a system that will be used in healthcare for many years. Wearable devices constantly collect massive volumes of data, including ambient conditions, body temperature, heart rate, and sweat levels. Data de-duplication using SHA2 helps eliminate repeated or duplicate data, optimizes storage space, and ensures that unique and relevant signals are processed for heat stroke detection. Many wearable gadgets store and manage data on the cloud. SHA-2 provides efficient cloud-based de-duplication by hashing data and eliminating duplicates before storage, saving storage costs and speeding up data retrieval and analysis. Wearable devices offer highly detailed, high-frequency data streams. Long bit-length hash values (256 bits or more) of SHA2 ensure that even minor differences in sensor data produce separate hash results. This ensures accurate de-duplication without mistakenly grouping relatively different sensor data as duplicates. De-duplicated data is secured using Advanced Encryption Standard (AES) to prevent unauthorized access and ensure data security. AES prohibits unauthorized access to critical heat stroke data by transforming sensor data into an unreadable format that can only be read with a key (figure 2).

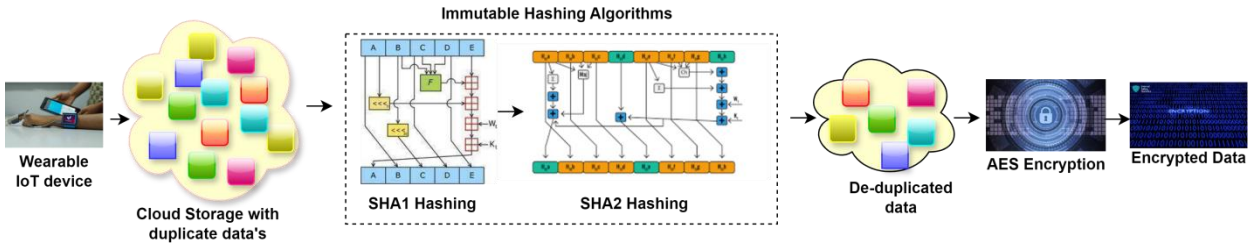

Figure 2: De-duplication process using Immutable Hashing Algorithms

SHA Process:

Step 1: Append padded bits:

Its length modulo 512 is comparable to 448 since the message is packed in this manner. Many zeros are required to achieve 448 mod 512; therefore, after adding a single 1-bit to the message's conclusion.

Step 2: Append length:

The output is the message's length represented in 64 bits. This step seeks to increase message length by 512 bits accurately.

Step 3: Parsing the message

Appending 64-bit blocks allows the padded message to be parsed into N 512-bit message blocks.(m^1,m^2 ,…, m,).

Step 4: Initialize Hash Value

The beginning hash value,H_0, consists of eight thirty-bit words in hexadecimal format.

Step 5: Get your messaging calendar ready. SHA256 uses a sixty-four thirty-bit word message schedule. The message schedule's words are labeled asZ_1,Z_2,,... Z_{63}Rotate right (RR) or shift right (SR).

$$Z_t=\begin{cases} M_t^t & 0<=t<=15 \\ \sigma_1^{256}(Z_{i-2}) + Z_{i-7} + \sigma_0^{256}(Z_{i-15}) + Z_{i-16} & 16\le t\le 63 \end{cases} \quad \text{-------- (1)}$$

Where,

$\sigma_1^{256}(Z_{i-2}) = ((Z_{i-2})\ RR\ 17) \oplus (Z_{i-2})\ RR\ 19) \oplus (Z_{i-2})$SR 10) ------ (2)

$\sigma_0^{256}(Z_{i-15}) = ((Z_{i-15})\ RR\ 7) \oplus (Z_{i-15})\ RR\ 18) \oplus (Z_{i-15})$SR 3) ------ (3)

Step 6: Initialize the eight working variables v1, v2, v3, v4, v5, v6, v7, and v8 with the (i-1) hash value for t=0 to 63: (ch-choose, maj-majority)

$T_1 = v8 + \sum_1^{256}(v5) + ch(v5v6.v7) + K_1^{256} + W_t$ --------- (4)

$T_2 = \sum_0^{256}(v1) + Maj(v1, v2, v3)$ ---------- (5)

V8 = V7, V7 = V6, V6 = V5, V5 = (v4+T_1), V4 = V3, V3 = V2, V2 = V1, V1 = $T_1 + T_2$

Where, $\sum_1^{256}(v5) = (v5\ RR\ 6) \oplus (v5\ RR\ 11) \oplus (v5\ RR\ 25)$ --------- (6)

$\sum_0^{256}(v1) = (v5\ RR\ 2) \oplus (v5\ RR\ 13) \oplus (\ v5\ RR\ 22)$ --------- (7)

$ch\big((v5v6.v7)\big) = (v5\text{^}v6) \oplus (\sim v5\text{^}v7)$ ---------- (8)

$Maj(v1, v2, v3) = (\ v1\text{^}v2) \oplus (v1\text{^}v3)\ \oplus (v2\text{^}v3)$ -------- (9)

Step 7: Output

Repetition of steps one through four N times results in a hash function:

$$H_0^N||H_1^N||H_2^N||H_3^N||H_4^N||H_5^N||H_6^N$$

To create the new hash H^i, first use all W^i input blocks and generate ω (63). Each input block of H^i is the sum of the corresponding input blocks of ω^i (63) and H^{i-1}. H^i (j) = H^{i-1} (j) + ω^i (63) (j), where + represents addition modulo 2^n. If W^i is the final message schedule, continue the operation. Otherwise, H^i = H is the final hash or digest for additional message blocks M^i.

Table 2: Comparison of SHA1 and SHA2

Feature	**SHA-1**	**SHA-2**
Hash Length	160 bits	256 bits (SHA-256), 512 bits (SHA-512)
Security	Weak; vulnerable to collisions	Strong; secure against collisions
Speed	Faster than SHA-2	Slower than SHA-1 but still efficient
Use Case	Legacy systems are not recommended for security-sensitive applications	Recommended for secure applications, widely used in modern systems
De-duplication	Fast for identifying duplicates but risks collisions	More reliable for de-duplication due to lower collision rates

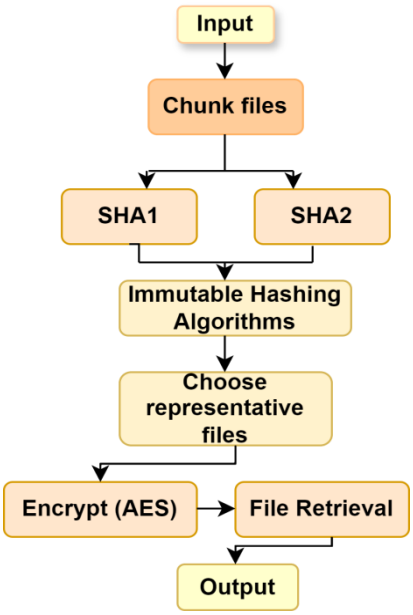

Figure 3: Workflow diagram of Immutable Hashing Algorithm

Figure 3 illustrates the process for the Hashing algorithm. First, when chunking the list of data, the initial data de-duplication technique performs the SHA algorithms and then combines the hash codes and files. Then, encrypted data is stored and accessed using the AES method for data security.

ALGORITHM 1: IMMUTABLE HASHING ALGORITHM

Input: De-duplication files (D), Chunk (C), Threshold (T), set of files {F1, F2… Fn}, a secret key K, and a public key PK.

Output: A list of unique files and their corresponding paths (U) and encrypted data

Step 1: Chunk the files

Generate a list of chunks with size C

Step 2: Compute Hash codes for SHA1with SHA2

Step 3: Combine Hash Codes

For each file, concatenate the hash codes from all chunks to form a hybrid hash code.

Step 4: Compare files

If the Hamming distance between the hash codes of two files is less than the threshold T, mark the files as potential duplicates and perform a byte-byte comparison.

Step 5: Choose Representative Files

For identical files, choose one file as the representative. Mark the other files as duplicates of the representative and Store the unique files.

Step 6: Encrypt the Files

Encrypt each unique file using AES with the secret key K and public key PK. Store the cipher-text files in a Big Data server. And Store the encrypted keys in a secure location.

Step 7: File Retrieval

To retrieve a file, decrypt its corresponding cipher-text key and file using Proxy-Re-Encryption and the secret key.

Output:

A list of unique files, their paths, and the encrypted data ready for storage and retrieval.

The files are first chunked in this algorithm, and two hash codes are created. Then, compare the files and perform a byte-by-byte comparison. One file will be chosen to represent each group of almost similar files; the other sets will be marked as duplicates. It preserves encrypted data and keys after using both public and private keys. The approach ensures that only authorized users may access the data by decoding the cipher text using the keys obtained during file acquisition.

IV RESULTS AND DISCUSSIONS

In this paper, we are implementing Immutable hashing Algorithms using Python. We are using SHA1 with SHA2 hashing algorithms to de-duplicate the heat stroke data. The proposed Immutable Hashing Algorithms perform well compared to other algorithms in de-duplication for heat stroke data. The hashing algorithm latency value is low compared to other algorithms. Throughput, Scalability, and Reliability values are higher in hashing algorithms than others.

Table 3: Comparison of various cloud metrics using multiple algorithms

Metrics/Algorithms	Latency/ms	Throughput (Data points/Sec)	Scalability/ million	Reliability
BLAKE 2	200 ms	120 /sec	5 /m	94.12
MD5	150 ms	140 /sec	1 /m	95.67
SHA1	100 ms	150 /sec	5 /m	96.54
SHA2	10 ms	900 /sec	10 /m	97.23
Immutable Hashing Algorithm	1 ms	1000 /sec	100 /m	98.23

Table 3 illustrates the comparison of BLAKE2, MD5, SHA1, SHA2, and Immutable Hashing Algorithms for data de-duplication and processing in a wearable heat-stroke detection system and highlights the performance in terms of latency, throughput, scalability, and reliability. The proposed method performs well compared to other existing algorithms.

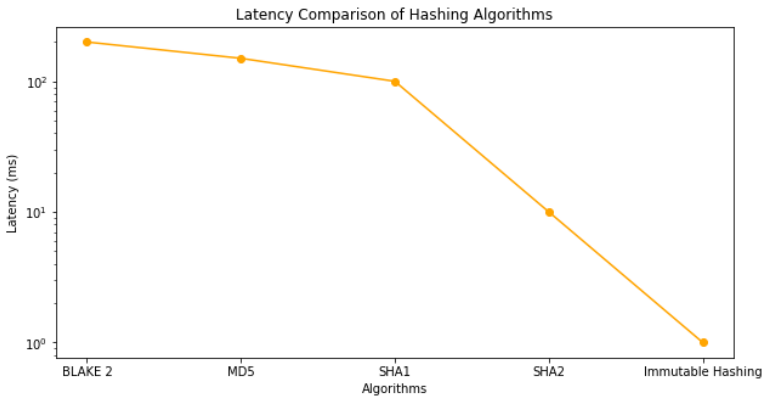

Figure 4: Latency comparison of various algorithms

Figure 4 shows the Latency Comparison of BLAKE2, MD5, SHA1, SHA2, and Immutable Hashing Algorithms. The proposed algorithm's latency is low compared to other existing algorithms. In the x-axis, various algorithms are marked, and the y-axis shows latency values.

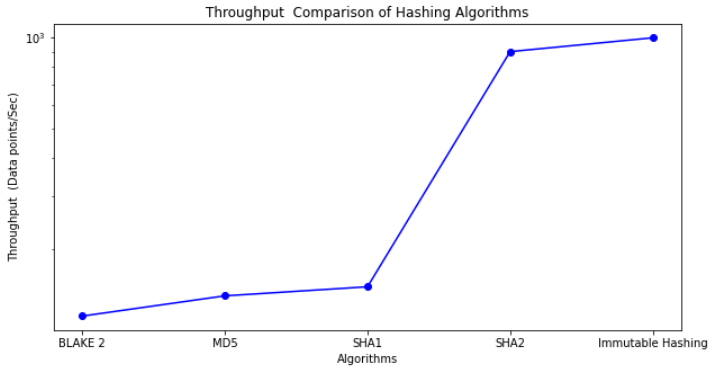

Figure 5: Throughput comparison of various algorithms

Figure 5 shows the Throughput Comparison of BLAKE2, MD5, SHA1, SHA2, and Immutable Hashing Algorithms. The proposed algorithm's throughput value is high compared to other existing algorithms. On the x-axis, various algorithms are mentioned, and on the y-axis, latency values are shown.

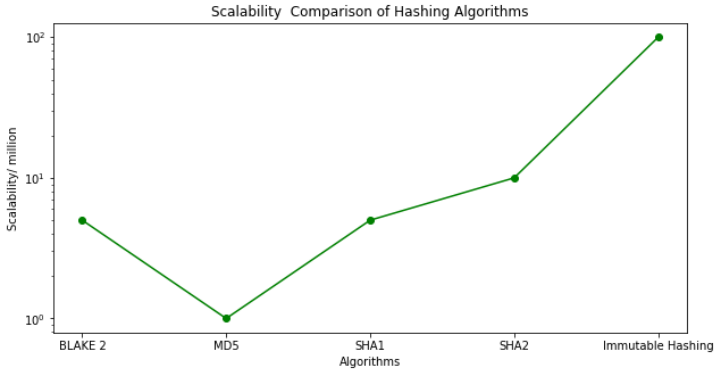

Figure 6: Scalability comparison of various algorithms

Figure 6 shows the Scalability Comparison of BLAKE2, MD5, SHA1, SHA2, and Immutable Hashing Algorithms. The proposed algorithm's scalability is high compared to other existing algorithms. In the x-axis, various algorithms are marked, and the y-axis shows latency values.

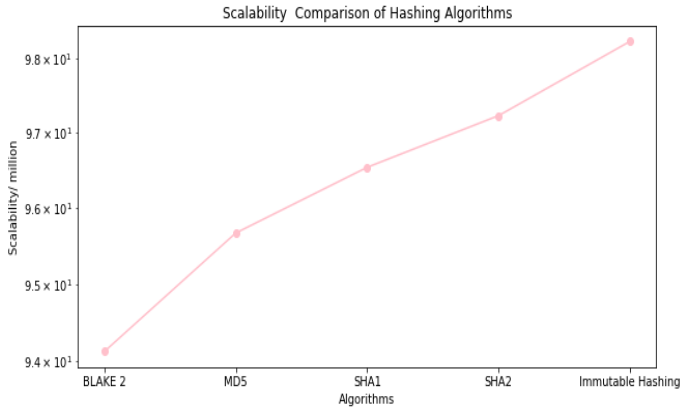

Figure 7: Reliability comparison of various algorithms

Figure 7 shows the Latency Comparison of BLAKE2, MD5, SHA1, SHA2, and Immutable Hashing Algorithms. The proposed algorithm's reliability value is high compared to other existing algorithms. In the x-axis, various algorithms are marked, and the y-axis shows latency values.

V CONCLUSION

In conclusion, implementing the Immutable Hashing Algorithm can significantly improve the performance and reliability of wearable heat-stroke detection systems and more effective and responsive healthcare solutions. In this research, we are using the Kaggle dataset for heat stroke detection. In big data, the data are repeatedly stored in duplicate and as redundant data when the wearable system sensor monitors the body temperature. So, the de-duplication process is used to find and clear the duplicate data and free the cloud storage in wearable heat stroke IoT-based systems for exemplary performance in heat stroke detection. This paper uses SHA1 with SHA2 hashing algorithms to decode heat stroke data. SHA1 performs well and fast, and SHA2 performs securely in de-duplication. Finally, AES prohibits unauthorized access to critical heat stroke data by transforming sensor data into an unreadable format that can only be read with a key. The result shows that immutable Hashing algorithms efficiently perform the data de-duplication process. It shows a high-reliability value (98.23 %), high throughput value (1000 sec), high scalability (100 m), and low latency (1m) value compared to existing algorithms. Future work could explore further optimizations and adaptations of these algorithms to enhance their effectiveness in various health monitoring scenarios.

REFERENCES

1. Majumder, S., Mondal, T., & Deen, M. J. (2017). Wearable sensors for remote health monitoring. *Sensors, 17*(1), 130.
2. Antonio, P. O., Rocio, C. M., Vicente, R., Carolina, B., & Boris, B. (2017, October). Heat stroke detection system based in IoT. In *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)* (pp. 1-6). IEEE.
3. Chen, S. T., Lin, S. S., Lan, C. W., & Hsu, H. Y. (2017). Design and development of a wearable device for heat stroke detection. *Sensors, 18*(1), 17.
4. El Ghazouani, M., Latifa, E. R., & El Khanboubi, Y. (2020). Efficient method based on Block-chain ensuring data integrity auditing with deduplication in cloud, International Journal of Interactive Multimedia and Artificial Intelligence, Vol. 6, No 3
5. El Khanboubi, Y., Hanoune, M., & El Ghazouani, M. (2021). A new data deletion scheme for a Block-chain-based de-duplication system in the cloud. *Int. J. Commun. Netw. Inf. Secur, 13*, 331-339.
6. Florian, M., Henningsen, S., Beaucamp, S., & Scheuermann, B. (2019, June). Erasing data from Block-chain nodes. In *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)* (pp. 367-376). IEEE.

7. Hintz, C., Presley, D. M., & Butler, C. R. (2024). Heat stroke burden and validity of wearable-derived core temperature estimation during elite military training. *The Physician and Sportsmedicine*, *52*(2), 154-159.

8. Hussain, I., Tahir, S., Humayun, M., Almufareh, M. F., Jhanjhi, N. Z., & Qamar, F. (2022, November). Health monitoring system using internet of things (iot) sensing for elderly people. In *2022 14th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)* (pp. 1-5). IEEE.

9. Javed, S., Ghazala, S., & Faseeha, U. (2020). Perspectives of Heat Stroke Shield: An IoT based Solution for the Detection and Preliminary Treatment of Heat Stroke. *Engineering, Technology & Applied Science Research*, *10*(2), 5576-5580.

10. Jaya, I., Maryanto, E., Bukit, A. V., & Zulkifli, M. (2018). Design And Development Of Early Heat Stroke Detection System In Military Cross Country Based On Iot. *JOURNAL ASRO*, *9*(2), 141-151.

11. Karmani, V., Chandio, A. A., Karmani, P., Chandio, M., & Korejo, I. A. (2019, July). Towards self-aware heatstroke early-warning system based on healthcare IoT. In *2019 Third World Conference on Smart Trends in Systems Security and Sustainablity (WorldS4)* (pp. 59-63). IEEE.

12. Lin, S. S., Lan, C. W., Hsu, H. Y., & Chen, S. T. (2018). Data analytics of a wearable device for heat stroke detection. *Sensors*, *18*(12), 4347.

13. Nor'en, M. S. I. B., & Chitturi, V. (2022, December). Wearable Technology for Early Detection of Hyperthermia Using Machine Learning. In *International Conference on Machine Learning, Image Processing, Network Security and Data Sciences* (pp. 252-263). Cham: Springer Nature Switzerland.

14. Ozdayi, M. S., Kantarcioglu, M., & Malin, B. (2020). Leveraging Block-chain for immutable logging and querying across multiple sites. *BMC Medical Genomics*, *13*, 1-7.

15. Pham, S., Yeap, D., Escalera, G., Basu, R., Wu, X., Kenyon, N. J., ... & Davis, C. E. (2020). Wearable sensor system to monitor physical activity and the physiological effects of heat exposure. *Sensors*, *20*(3), 855.

16. Rahardja, U., Hidayanto, A. N., Lutfiani, N., Febiani, D. A., & Aini, Q. (2021). Immutability of distributed hash model on Block-chain node storage. *Sci. J. Informatics*, *8*(1), 137-143.

17. Rahman, F. (2024). Big Data Analytics Based Data Driven Public Health Care System For Heart Disease Detection. *South Eastern European Journal of Public Health*, 321-326.

18. Rajkumar, K., & Dhanakoti, V. (2022). Fuzzy-Dedup: A secure deduplication model using cosine based Fuzzy interference system in cloud application. *Journal of Intelligent & Fuzzy Systems*, *43*(3), 2819-2832.

19. Son, T. W., Ramli, D. A., & Abd Aziz, A. (2021). Wearable heat stroke detection system in IoT-based Environment. *Procedia Computer Science*, *192*, 3686-3695.

20. Srivastava, S. K., Maakar, S. K., Singh, H. R., Srivastava, D., & Kantha, P. (2023). Supervision of Worldwide Healthcare through an IoT-Based System. In *Intelligent Internet of Things for Smart Healthcare Systems* (pp. 113-131). CRC Press.

21. Tuli, S., Basumatary, N., Gill, S. S., Kahani, M., Arya, R. C., Wander, G. S., & Buyya, R. (2020). HealthFog: An ensemble deep learning based Smart Healthcare System for Automatic Diagnosis of Heart Diseases in integrated IoT and fog computing environments. *Future Generation Computer Systems*, *104*, 187-200.

22. Vyas, A., Abimannan, S., & Hwang, R. H. (2021). Sensitive Healthcare Data: Privacy and Security Issues and Proposed Solutions. *Emerging Technologies for Healthcare: Internet of Things and Deep Learning Models*, 93-127.

23. Wan, J., AAH Al-awlaqi, M., Li, M., O'Grady, M., Gu, X., Wang, J., & Cao, N. (2018). Wearable IoT enabled real-time health monitoring system. *EURASIP Journal on Wireless Communications and Networking*, *2018*(1), 1-10.

24. Wijaya N.H., Fauzi F.A., Helmy E.T., Nguyen P.T., and Atmoko R.A (2020) "The esign of Heart Rate etector and Body Temperature Measurement evice Using TMega16," Journal of Robotics and Control (JRC) 1:40–43.

25. Yin, L. K., Yogarayan, S., Razak, A., Fatimah, S., Bukar, U. A., & Sayeed, M. S. (2024). Heat stroke prediction: a perspective from the internet of things and machine learning approach. *International Journal of Electrical & Computer Engineering (2088-8708)*, *14*(3).

26. Mohamad Emad Bitar and Dr. V. Sujatha, (2024), "Combination Of Fused Machine Learning And Cascaded Levy Flight Optimization In Heat Stroke Prediction", Nanotechnology Perceptions Vol.20, S14 (2024).