Bike computer forensics: An efficient and robust method for FIT file recovery

By:
Kwangkeun Song, Dongbin Oh

From the proceedings of
The Digital Forensic Research Conference
DFRWS APAC 2023
Oct 17-20, 2023

Forensic Science International: Digital Investigation 46 (2023) 301606

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

journal homepage: www.elsevier.com/locate/fsidi

DFRWS 2023 APAC - Proceedings of the Third Annual DFRWS APAC

Bike computer forensics: An efficient and robust method for FIT file recovery

Kwangkeun Song [a,*], Dongbin Oh [b]

[a] Digital Forensics Section, Cyber Investigation Division, Seoul Metropolitan Police Agency, Seoul, South Korea
[b] Korea University School of Cybersecurity, Seoul, South Korea

ARTICLE INFO

Keywords:
Bike computer forensics
FIT file recovery
Ride data analysis
Sliding window pattern match

ABSTRACT

The popularity of bike computer devices has grown in recent years. These devices generate a wealth of data in the form of Flexible and Interoperable Data Transfer (FIT) files, which can be used to store fitness related data efficiently. However, the recovery of corrupted FIT files remains a significant challenge due to their inherent structure. The format relies on a chain of messages stored sequentially, with each message referencing previous data to parse the subsequent record. As a result, the recovery of data situated between corrupted portions becomes notably challenging. This study introduces an efficient, and robust method for dense recovery of corrupted files. Our approach combines multiple phases of data carving techniques to maximize data recovery. By employing this method, investigators can effectively access crucial information including accident reconstruction, and criminal activities. The proposed methods demonstrate higher recovery rate through the proof-of-concept and real-world experiments, proving its utility and reliability in the field of digital forensics.

1. Introduction

A bike computer is a portable electronic device that can be mounted to a bicycle to measure the cycle related metrics and the ride itself. The common measurements displayed on a bike computer are time, distance, power, heart rate, cadence, temperature, and so on.

In the past, only professional cyclists tracked their riding metrics for training, skill improvement, and correcting factors such as riding posture. However, modern bike computers provide rich information in real-time, making them increasingly popular among ordinary individuals for safety purposes and self-motivation to maintain an active exercise routine.

Ride data received by the various sensors is saved in the form of FIT file format. The format was developed by Garmin. It has flexibility and interoperability for storing and transferring fitness-related data. The players stated in Mordor Intelligence (2023) and Custom Market Insights (2023) are Garmin, Lezyne Inc., Bryton Inc., CatEye Co.Ltd., Wahoo Fitness and so on. We confirmed from their websites and technical blogs that majority of them also supports the format. While it is widely adopted and used in the fitness industry, FIT files can occasionally become corrupted due to following issues:

- Lower battery power
- Device crashes (hardware or software)
- Data transmission problems

The official FIT cookbook website and the Strava community also acknowledge the possibility of data corruption. Users frequently encounter corruption and seek assistance in repairing them through Garmin forums. Moreover, sensor data is conveyed in a chain-like message format. These messages are linked together and stored sequentially. They rely on references to previous data for parsing the next record. In such a connected structure, restoring the remaining data becomes challenging when encountering corrupted portions within the sequence as they reference the wrong offset. In addition, in situations involving bike accidents, the ability to recover complete data from damaged files becomes crucial.

This research aims to address these challenges and propose methods for data recovery in bike computers, with a specific focus on record-level recovery techniques. By employing these methods, it becomes possible to reconstruct the sequence of events leading up to a bike accident or analyze potential evidence related to criminal activities. The findings of this study can contribute to investigation of incidents by leveraging the available data from fitness devices.

---

* Corresponding author.
E-mail address: fkilla8210@gmail.com (K. Song).

https://doi.org/10.1016/j.fsidi.2023.301606

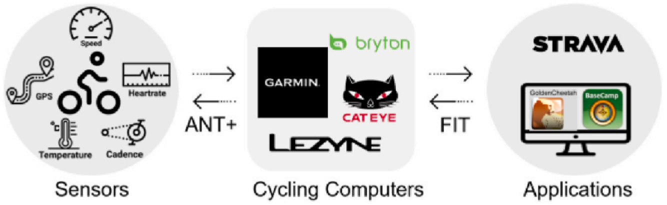


Fig. 1. Bike computer with sensors and services.

This paper is structured as follows: Section 2 provides a detailed explanation of the protocols utilized by bike computers and how they interact with sensors and services. Section 3 presents a review of previous research on fitness device forensics. In Section 4, we discuss the relationships between messages and the challenges of recovery. Section 5 and 6 outline our proposed recovery method and evaluate its effectiveness in recovering any remaining data from the corrupted FIT files. Finally, in Section 7, we conclude the paper.

2. Background

2.1. ANT and ANT+

Bike computers can measure more metrics if there are more sensors attached to the bike. A protocol is required to communicate between the device and the various sensors. The mainly used one in bike computers is Advanced and Adaptive Network Technology (ANT). ANT is a wireless sensor network protocol running in the 2.4 GHz ISM band. It is used in various fields such as sports, health care devices, and home appliances.

ANT+ is a set of mutually agreed-upon definitions for what the information sent over ANT represents. It is widely adopted in fitness devices due to its utilization of proven ultra-low power that consumes less power and provides high reliability. Additionally, ANT + allows for simultaneous connection of sensor accessories to multiple devices, enabling data transmission at the same time. In contrast, Bluetooth typically supports the pairing of multiple accessories with a single device.

2.2. FIT protocol

The Flexible and Interoperable Data Transfer (FIT) protocol is a data format utilized for storing and sharing health device-generated data. The format allows for various interpretations and processing methods, such as visualizing with graphs or mapping GPS data, so to determine the distance and locations traveled by a user. The FIT protocol is designed with scalability, accommodating the development of diverse sensors.

The format is defined in the FIT Software Development Kit (FIT SDK), so fitness device manufacturers can utilize it to add or incorporate data to suit their individual needs. Fig. 1 shows the general process of interoperating fitness data. The bike computer collects various sensor data through ANT+ and generates data in the FIT format.

While the bike computer provides the ability to view the workout data, its relatively small display makes it difficult to see the analyzed data in detail. Therefore, online platforms such as Strava can be utilized to better manage fitness data and participate in communities by competing with other users.

3. Related work

Previous studies have focused on various digital forensic aspects of fitness tracking devices such as data extraction methods, data analysis, forensic tools, privacy concerns, and security considerations.

Table 1
FIT file recovery tools.

Tools	Download URL
GOTOES utility for Strava	https://gotoes.org/strava/Combine_GPX_TCX_FIT_Files.php
FIT Viewer Tool by RUNALYZE	https://runalyze.com/tool/fit-viewer
(FIT SDK) Activity file repair tool	https://developer.garmin.com/fit/cookbook/activity-file-repair-tool/
FIT File Tools	https://fitfiletools.com/
Garmin Fit file conversion and repair	http://garmin.stevegordon.co.uk/

3.1. Data extraction and analysis

Kim et al. (2023) established a forensic model for wearable devices that utilized both physical and logical interfaces. The model was then applied to devices from Samsung, Apple, and Garmin. The study concluded that it was effective for forensic analysis of these devices. Á. MacDermott et al. (2019) researched Garmin, Fitbit, and HETP wearable devices to identify files of interest and their locations. They also stated the possibility of recovering FIT files that cannot be viewed with common forensic tools. Kang et al. (2020) proposed the method to extract and analyze the user's health data. Specifically, their research focused on analyzing data that is connected to or synchronized with Android smartphones.

3.2. Forensic tool concerns

Dawson and Akinbi (2021) analyzed TomTom smartwatch, mobile app, and Bluetooth event logs under the investigative scenario. They concluded that commercial and traditional forensic tools had to be compensated with the use of non-forensic tools. Hantke and Dewald (2020) performed forensic analysis on three popular fitness trackers - Xiaomi Mi Band 2, Fitbit Charge 2 and Huawei Band 2 Pro. Additionally, they developed an open-source tool that enables forensically sound analysis of these devices.

3.3. Security and privacy concerns

Hossein et al. (2017) involved a comprehensive security analysis of 17 popular fitness trackers available in the market. Based on their findings, the researchers recommended that manufacturers should implement secure hardware and software development practices and integrate privacy and security measures into their existing processes. Hutchinson et al. (2022) focused their study on three wearable device applications - Amazon Halo, Garmin Connect, and Mobvoi. During their analysis, they were able to recover user profiles and health-related data from these applications. In addition, in their study, a partial voice recording was recovered from Amazon Halo.

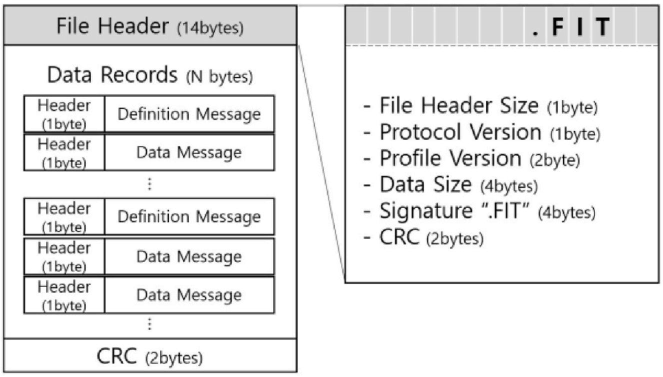


Fig. 2. FIT file layout.

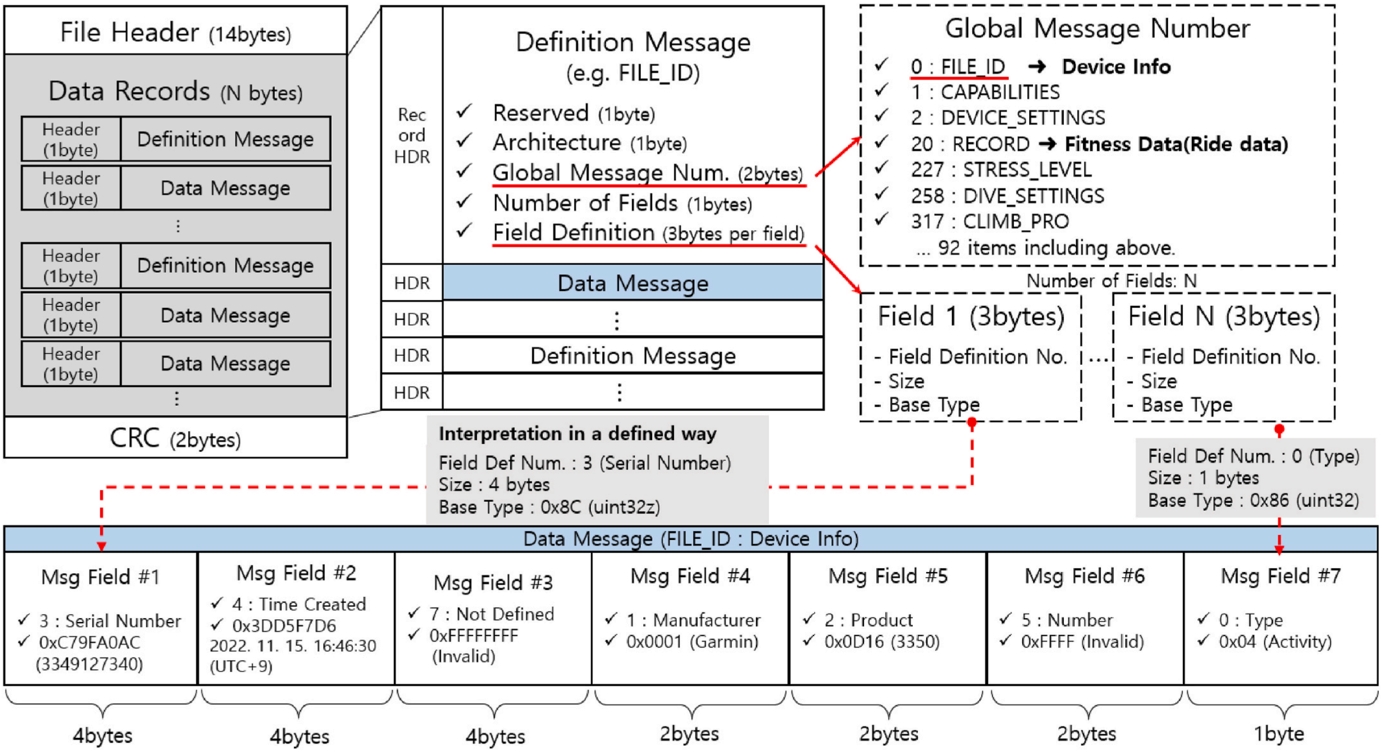


Fig. 3. Process of decoding a data message.

Despite the extensive research conducted on fitness device forensics, none of the previous studies have thoroughly investigated the recovery of a specific file format, FIT, that is commonly used in bike computers.

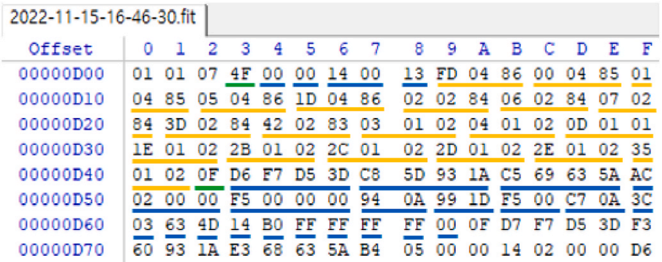
As shown in Table 1, we listed publicly available FIT repair tools found online. These tools claim to have repair functionality on their websites or forums, and we will compare their capabilities with our proposed recovery method in Section 6.

4. FIT file structure

4.1. FIT layout

A FIT file consists of three major parts according to official FIT reference: a file header, data records, and a 2-byte CRC value. The file header, shown in Fig. 2, contains the signature '.FIT' and the size of the data records section. The header also includes CRC to check the integrity of the header part.

The data records section contains both definition messages and data messages that are save next to one another. A definition message specifies how to interpret upcoming data messages, whereas a data message is a piece of measured ride data itself and is encoded in the format described in the preceding definition messages. A CRC is also contained at the end of file.



```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000D00  01 01 07 4F 00 00 14 00  13 FD 04 86 00 04 85 01
00000D10  04 85 05 04 86 1D 04 86  02 02 84 06 02 84 07 02
00000D20  84 3D 02 84 42 02 83 03  01 02 04 01 02 0D 01 01
00000D30  1E 01 02 2B 01 02 2C 01  02 2D 01 02 2E 01 02 35
00000D40  01 02 0F D6 F7 D5 3D C8  5D 93 1A C5 69 63 5A AC
00000D50  02 00 00 F5 00 00 00 94  0A 99 1D F5 00 C7 0A 3C
00000D60  03 63 4D 14 B0 FF FF FF  FF 00 0F D7 F7 D5 3D F3
00000D70  60 93 1A E3 68 63 5A B4  05 00 00 14 02 00 00 D6
```

Fig. 4. An example of ride data (Global Message: Record).

4.2. Definition message

A definition message is composed of a fixed 5-byte length value (record header excluded), and a variable-length field definition value, as shown in Fig. 3. It plays an important role in interpreting the actual workout data and is always stored ahead of the data message in the FIT file, followed by one or more data messages. The definition message includes important factors such as the global message number, number of fields, and field definition.

The global message number represents a large category that indicates the type of data conveyed by the data message. Each number corresponds to a different category of items, with a total of 92 entries defined in the FIT SDK. For example, the number '0' represents 'FILE_ID', indicating the device information of the bike computer, while the number '20' (Record) represents workout records.

The number of fields in the definition message indicates the count of fields that should be decoded in the data message.

Lastly, the variable-length field definition gives a detailed method to interpret the actual data. The elements of field definition are as follows:

- Field definition number (1 byte)
- Field size (1 byte)
- Base type (1 byte)

Like the global message number, field definition also has a list of specific field names defined according to their numbers. Therefore, the subcategory of each field can be determined using this number.

The field size refers to the length of each field, and the base type specifies the data type (unsigned int, string, etc.) of a field for the FIT decoder to handle unknown or invalid data. As a result, the upcoming data messages can be decoded based on the factors defined in the field definition.

All global message numbers, field definitions, and base types are defined in the FIT SDK. Fig. 3 illustrates how both messages get connected and used together to decode the device information.

Table 2
Analysis results of ride data (Global Message: Record).

Hex value	Description	Hex value	Description
4F	0100 (upper 4 bits): Definition Message 1111 (lower 4 bits): Local Message Number	0F	0000: Data Message 1111: Local Message Number
00	0: Reserved		
00	0: Little-endian		
1400	0x14(20): Global Message Number ➔ RECORD		
13	0x13(19): Number of fields		

Hex value	Field Definition	Size (bytes)	Base Type	Hex value	Description
FD0486	Timestamp	4	0x86 (uint32)	D6F7D53D	2022-11-15 16:46:30 (UTC+9)
000485	Position Latitude	4	0x85 (sint32)	C85D931A	37.372007 (°)
010485	Position Longitude	4	0x85 (sint32)	C569635A	127.10859 (°)
050486	Distance	4	0x86 (uint32)	AC020000	6.84 (m)
1D0486	Accumulated Power	4	0x86 (uint32)	F5000000	245 (watts)
020284	Altitude	2	0x84 (uint16)	940A	41.60 (m)
060284	Speed	2	0x84 (uint16)	991D	7.577 (m/s)
070284	Power	2	0x84 (uint16)	F500	245 (watts)
3D0284	Undefined	2	0x84 (uint16)	C70A	Field_61 (Undefined): 2759
420283	Undefined	2	0x83 (sint16)	3C03	Field_66 (Undefined): 828
030102	Heart Rate	1	0x02 (uint8)	63	99 (bpm)
040102	Cadence	1	0x02 (uint8)	4D	77 (rpm)
0D0101	Temperature	1	0x01 (sint8)	14	20 (°C)
1E0102	Left Right Balance	1	0x02 (uint8)	B0	176
2B0102	Left Torque Effectiveness	1	0x02 (uint8)	FF	Invalid value (%)
2C0102	Right Torque Effectiveness	1	0x02 (uint8)	FF	Invalid value (%)
2D0102	Left Pedal Smoothness	1	0x02 (uint8)	FF	Invalid value (%)
2E0102	Right Pedal Smoothness	1	0x02 (uint8)	FF	Invalid value (%)
350102	Fractional Cadence	1	0x02 (uint8)	00	0.0 (rpm)

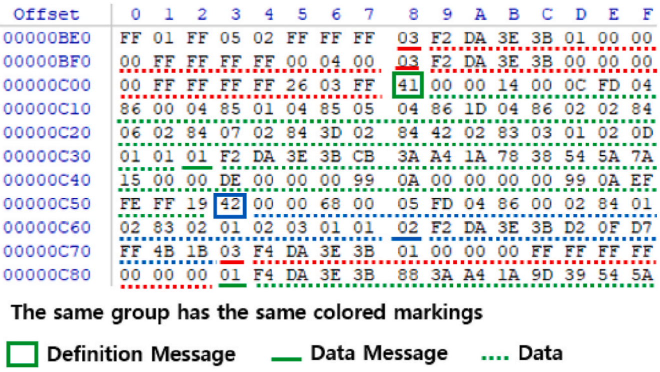
4.3. Data message

A Data message consists of several data fields. Since each field name related to the data is already specified in the definition message, no other elements are stored in the data message, except for the measured data according to the global message number.

An example of ride data, global message number 20 (Record), is parsed as shown in Fig. 4, and the description of each data is detailed in Table 2.

4.4. Challenges of FIT file structure

Definition messages and data messages are associated with the local message numbers. The local message number which is the four least significant bits of the record header, is used to seek the corresponding definition message to interpret the ride data.



```
Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000BE0  FF 01 FF 05 02 FF FF FF  03 F2 DA 3E 3B 01 00 00
00000BF0  00 FF FF FF FF 00 04 00  03 F2 DA 3E 3B 00 00 00
00000C00  00 FF FF FF FF 26 03 FF  41 00 00 14 00 0C FD 04
00000C10  86 00 04 85 01 04 85 05  04 86 1D 04 86 02 02 84
00000C20  06 02 84 07 02 84 3D 02  84 42 02 83 03 01 02 0D
00000C30  01 01 01 F2 DA 3E 3B CB  3A A4 1A 78 38 54 5A 7A
00000C40  15 00 00 DE 00 00 00 99  0A 00 00 00 00 99 0A EF
00000C50  FE FF 19 42 00 00 68 00  05 FD 04 86 00 02 84 01
00000C60  02 83 02 01 02 03 01 01  02 F2 DA 3E 3B D2 0F D7
00000C70  FF 4B 1B 03 F4 DA 3E 3B  01 00 00 00 FF FF FF FF
00000C80  00 00 00 01 F4 DA 3E 3B  88 3A A4 1A 9D 39 54 5A
```

Fig. 5. Relationships between messages.

Regardless of the message type, they are stored next to each other. As shown in Fig. 5, the first green data message is saved immediately after the definition message. Furthermore, it is common for a series of data messages to be sequentially arranged, like the pattern of the first and second red data messages.

However, data messages are not necessarily stored one after the other.

In some cases, they are in the middle between other messages, such as the third red data message. Therefore, even though the data message isn't directly following the definition message, it can still refer to the local message number in the header to find its corresponding definition message and parse the data as a result.

When intermediate data is damaged, the interpretation of the remaining ride data becomes challenging. This occurs because the damaged portion causes the data to refer to incorrect offsets, leading to a loss of remaining data and making it difficult to extract accurate information.

Data corruption may occur due to various factors, including the impact of bike accidents, unexpected events such as system freeze, or more as stated in Section 1.

5. Proposed recovery method

In the FIT file, the file header contains the signature as well as the size of the entire data record area. Therefore, it is necessary to carve out the file first, and then restore the data messages separately using the proposed methods.

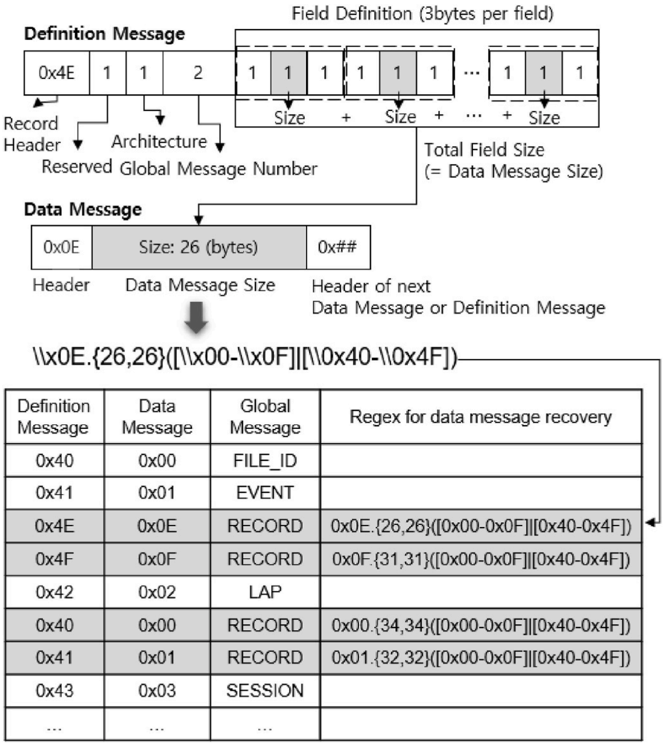
5.1. Phase 1: definition message search

To ensure accurate parsing of data messages, the initial step involves identifying the definition messages' position within the file. This allows for the parser to proceed to the next closest definition message's location in the event of a fault encountered during message parsing. When a data message cannot be decoded correctly, the parser can use the offset of the next nearest definition message as a new starting point to resume parsing subsequent data. Thus, it enables the parser to handle corrupted messages effectively and ensures uninterrupted data parsing.

The regular expression that can be used to search for definition messages is "[\x40-\x4F]\x00\x00([\x00-\xFF]\x00|[\x00-\x77]\x01)[\x01-\x64]".

Definition messages are not consistently stored closely together, there may be some distance between them. Simply skipping over to the next definition message's offset whenever decoding fails is not an ideal solution, as it would omit numerous normal data messages situated in between. Moreover, cycle workout data are generated at a rate of one per second. Missing a small portion of data could result in the loss of numerous accident-related information.

As a result of this structural limitation, dense restoration operations



Definition Message	Data Message	Global Message	Regex for data message recovery
0x40	0x00	FILE_ID	
0x41	0x01	EVENT	
0x4E	0x0E	RECORD	0x0E.{26,26}([0x00-0x0F]|[0x40-0x4F])
0x4F	0x0F	RECORD	0x0F.{31,31}([0x00-0x0F]|[0x40-0x4F])
0x42	0x02	LAP	
0x40	0x00	RECORD	0x00.{34,34}([0x00-0x0F]|[0x40-0x4F])
0x41	0x01	RECORD	0x01.{32,32}([0x00-0x0F]|[0x40-0x4F])
0x43	0x03	SESSION	
…	…	…	

Fig. 6. Generate an approximate regex to find data messages.

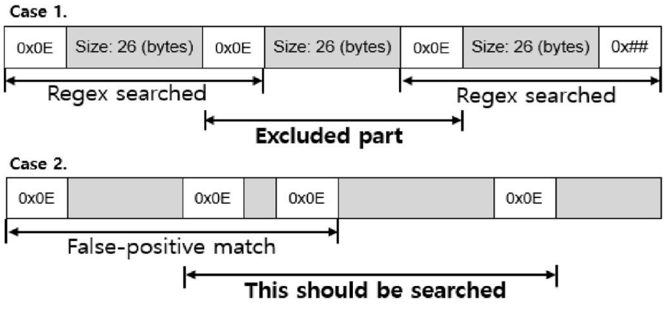


Fig. 7. Issues of general regex search.

will be required for the data messages to be recovered in between.

5.2. Phase 2: generate regex to search data messages

Each data message has its variable length and is very different depending on how they are defined in the related definition message. For this reason, making a precise regex, only targeting for recovery area, to search data messages is another challenge. The simplest and clearest regex is to combine the three following elements:

- Data message header (1 byte)
- Size of the data message (variable length)
- Header of next definition or data message (1 byte)

The size of each field can be obtained from its definition message, and by adding these individual sizes, the necessary part of the regex can be determined as illustrated in Fig. 6.

5.3. Phase 3: sliding window pattern match

The recovery approach outlined in Section 5.2, presents another challenge. Fig. 7 demonstrates two instances where simplified expressions failed to identify certain data messages. In the first case, the expression excluded the data message in between, because the regex contained the header of the next message. Additionally, in the second case, false positives may occur, making it difficult to conduct an exact search.

None of the cases fully detects the messages during the scan of the recovery area. It's because once the regex algorithm has found a match, the matched data is not considered for further matches.

To overcome these issues, it is necessary for the algorithm to continuously search once a match is encountered. As shown in Fig. 8, sliding window pattern match involves moving a fixed-size window through the match checking for another match for each step. Thus, it can narrow down the false positives as well as produce better recovery rate than using a typical regex search.

By implementing these approaches, it is possible to minimize the unrestorable area as shown in Fig. 9.

The flowchart shown in Fig. 10, outlines our methodology described above.

6. Experiment and results

6.1. Proof-of-concept experiment

We conducted a proof-of-concept research to investigate the effects of data loss in a FIT file, acquired from a Garmin Edge830 device. To obtain a FIT file, forensic imaging of the device was performed by Forensic Falcon, and then we extracted a file from the obtained e01 image. Subsequently, a deliberate deletion of a specific area in the file was carried out, assuming the loss of multiple consecutive data messages due to unexpected events.

The red boxes shown in Figs. 11 and 12 indicate the deleted regions that are removed. Each highlighted block represents a data message. Fig. 11 contains a total of 8 message blocks. The first block is partially damaged at the end, six are entirely deleted, and the last block is damaged only in the front.

Also, we conducted an additional experiment in our research, considering the data chunks are processed at the sector level as the

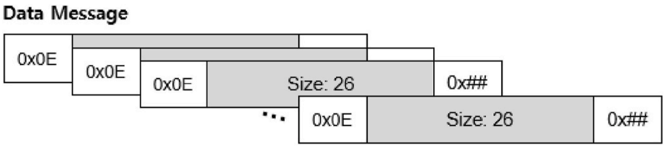


Fig. 8. Sliding window pattern match.

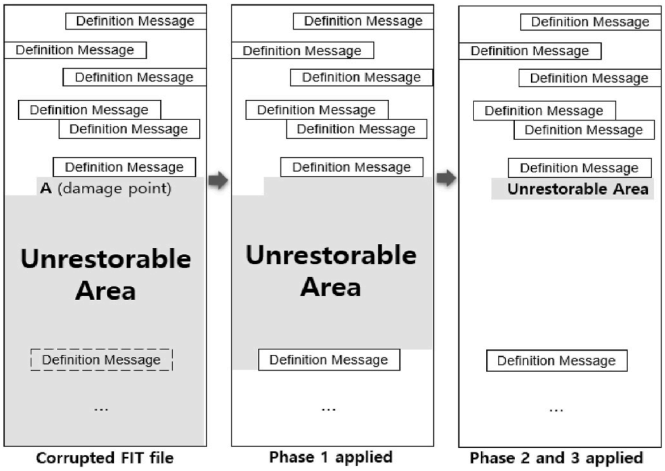


Fig. 9. Minimized unrestorable area by phase.

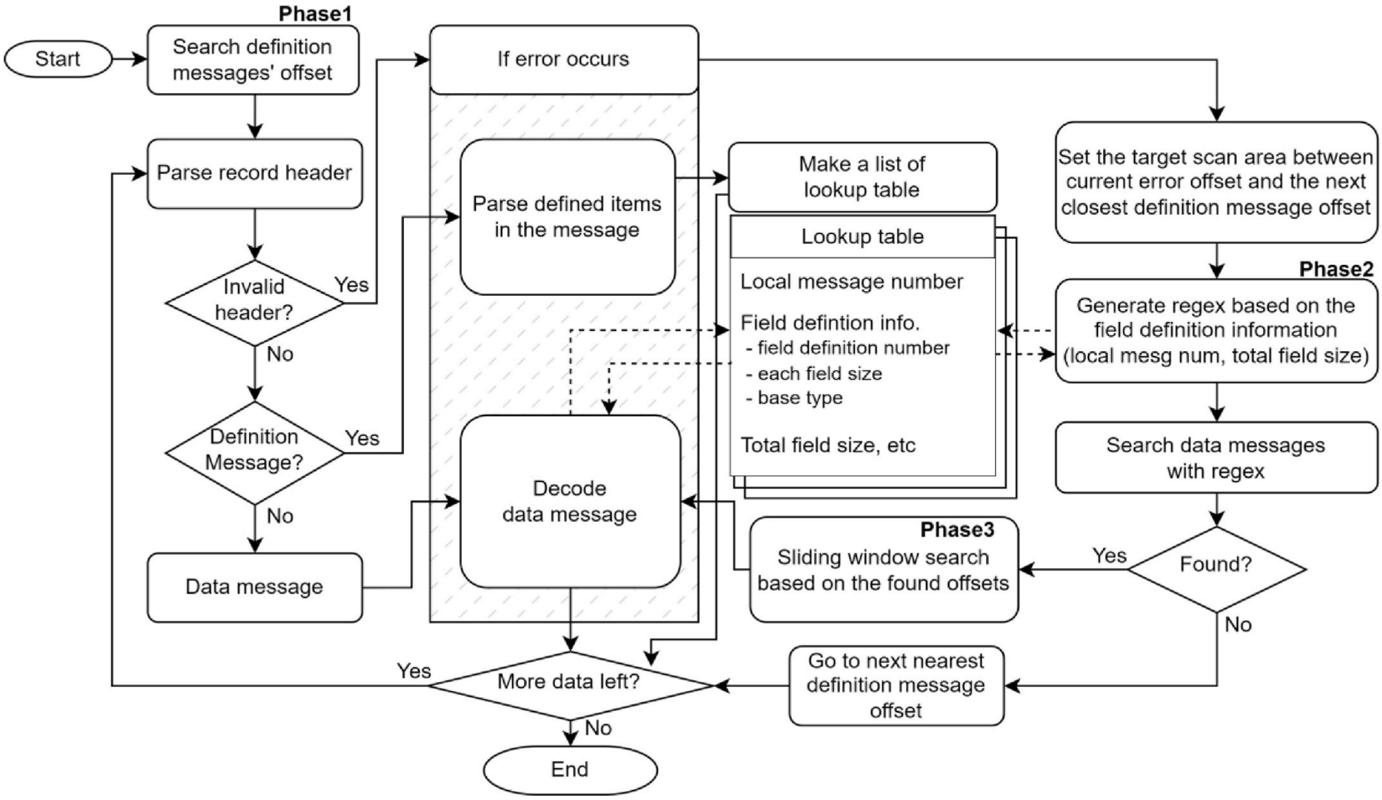
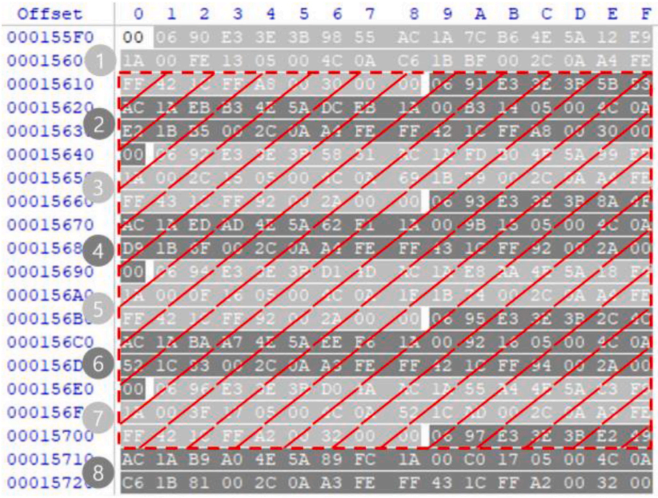
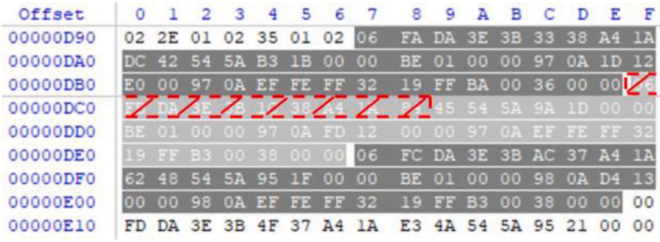


Fig. 10. Flowchart of our proposed method.



```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
000155F0   00 06 90 E3 3E 3B 98 55  AC 1A 7C B6 4E 5A 12 E9
0001560 ①  1A 00 FE 13 05 00 4C 0A  C6 1B BF 00 2C 0A A4 FE
00015610   FF 42 1C FF A8 00 30 00  00 06 91 E3 3E 3B 5B 53
0001562 ②  AC 1A EB B3 4E 5A DC EB  1A 00 B3 14 05 00 4C 0A
0001563    E2 1B B5 00 2C 0A A4 FE  FF 42 1C FF A8 00 30 00
00015640   00 06 92 E3 3E 3B 58 31  AC 1A FD B0 4E 5A 99 FE
0001565 ③  1A 00 2C 15 05 00 4C 0A  69 1B 79 00 2C 0A A4 FE
0001566    FF 43 1C FF 92 00 2A 00  00 06 93 E3 3E 3B 8A 4E
00015670   AC 1A ED AD 4E 5A 62 E1  1A 00 9B 15 05 00 4C 0A
0001568 ④  D9 1B 6F 00 2C 0A A4 FE  FF 43 1C FF 92 00 2A 00
00015690   00 06 94 E3 3E 3B D1 4D  AC 1A E8 AA 4E 5A 18 FE
000156A ⑤  1A 00 0F 16 05 00 4C 0A  1F 1B 74 00 2C 0A A4 FE
000156B    FF 42 1C FF 92 00 2A 00  00 06 95 E3 3E 3B 2C 4C
000156C0   AC 1A BA A7 4E 5A EE E6  1A 00 92 16 05 00 4C 0A
000156D ⑥  52 1C 33 00 2C 0A A3 FE  FF 42 1C FF 94 00 2A 00
000156E0   00 06 96 E3 3E 3B D0 1A  AC 1A 55 A4 4E 5A E3 FE
000156F ⑦  1A 00 3F 17 05 00 4C 0A  52 1C AD 00 2C 0A A3 FE
00015700   FF 42 1C FF A2 00 32 00  00 06 97 E3 3E 3B E2 49
0001571 ⑧  AC 1A B9 A0 4E 5A 89 FC  1A 00 C0 17 05 00 4C 0A
0001572    C6 1B 81 00 2C 0A A3 FE  FF 43 1C FF A2 00 32 00
```

Fig. 11. Case 1: 8 records are damaged.



```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000D90   02 2E 01 02 35 01 02 06  FA DA 3E 3B 33 38 A4 1A
00000DA0   DC 42 54 5A B3 1B 00 00  BE 01 00 00 97 0A 1D 12
00000DB0   E0 00 97 0A EF FE FF 32  19 FF BA 00 36 00 00 06
00000DC0   FD DA 3E 3B 1C 38 A4 1A  84 45 54 5A 9A 1D 00 00
00000DD0   BE 01 00 00 97 0A FD 12  00 00 97 0A EF FE FF 32
00000DE0   19 FF B3 00 38 00 00 06  FC DA 3E 3B AC 37 A4 1A
00000DF0   62 48 54 5A 95 1F 00 00  BE 01 00 00 98 0A D4 13
00000E00   00 00 98 0A EF FE FF 32  19 FF B3 00 38 00 00 00
00000E10   FD DA 3E 3B 4F 37 A4 1A  E3 4A 54 5A 95 21 00 00
```

Fig. 12. Case 2: One record is partially deleted in the front.

Table 3
Comparison of ride records recovered by phase.

Case	Total records	Parse mode	Recovered records by each phase		
			Phase 1	Phase 1+2	Phase 1+2+3
Case 1: 6 data messages deleted, 2 partially damaged	4456	2123	3120	3650	4449
Case 2: 1 data message partially damaged	4456	27	4270	4342	4455
Case 3: 1 sector deleted (11 data messages deleted, 2 partially damaged)	4456	2482	2566	3335	4443
Case 4: 1 sector deleted (1 definition message deleted, 11 data messages deleted, 1 partially damaged)	4456	30	43	160	763

device uses a FAT32 file system. In the experiment, we removed a randomly selected sector regardless of whether it contained definition messages or not.

We applied the method and evaluated the recovery rate at each step. Since a timestamp is an artifact that serves as an objective reference of all records, we evaluated the result by counting the recovered data messages that contained valid timestamps.

The results of the recovered data, detailed in Table 3, demonstrate that all data messages including the partially damaged portion at the front were properly recovered. We also observed that the number of recovered records is approximately 1,000 higher when the phase 1 method is applied compared to the parse mode. This indicates that the loss of just one byte of data could result in the loss of roughly 1,000 records of important criminal activity, which amounts to 17 minutes

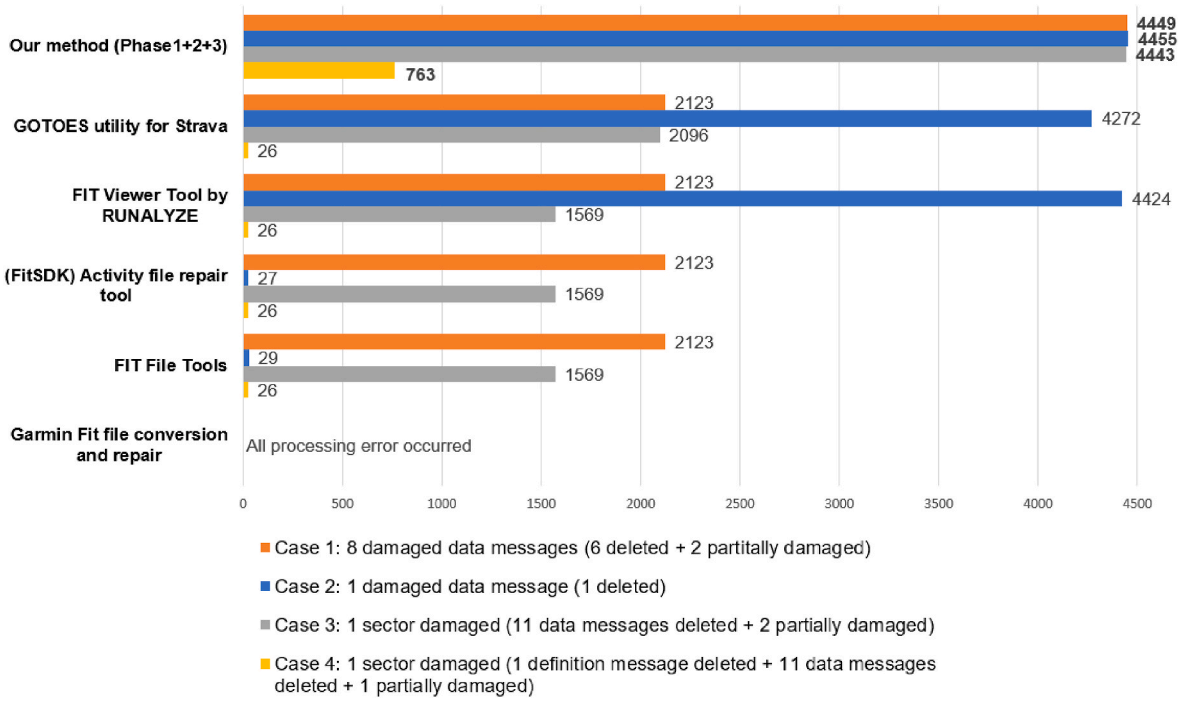


Fig. 13. Comparison of recovered ride records by our method and other recovery tools.

worth of data.

One limitation of the parse mode is its inability to decode messages when encountering failures. Therefore, the result of parse mode between Case 2 and 3 implies which portion of the file is corrupted.

Our results, as shown in Fig. 13, produced higher recovery rates with valid ride data compared to recovery tools as listed in Table 1. Additionally, we provide all scripts publicly available on Github[1] for others to use and build upon.

As for case 1, four tools marked the same records due to their reliance on the parsing process, which can cause the process to stop if errors are encountered. However, in the result of case 2, notable recovery outputs were observed by the 'GOTOES utility for Strava' and 'FIT Viewer Tool by RUNALYZE'. These two tools utilized the phase 1-based method, resulting in higher outcomes.

When the definition message is deleted, as observed in case 4, the recovery rate significantly decreases due to the inability to interpret the corresponding data messages. This means without the presence of the definition message, the proper reconstruction and interpretation of the associated data become another challenging.

6.2. Real-world FIT recovery experiment

In addition to conducting proof-of-concept experiments, we performed recovery tests on several corrupted FIT files[2] downloaded from the Garmin forum. These files were uploaded by users seeking assistance in repairing their files. The inclusion of real-world corrupted files allowed us to assess the practical applicability and effectiveness of the proposed recovery methods. The evaluation includes using a total of 6 tools, including our own. Our methods demonstrated the highest data recovery rate, surpassing other tools in terms of overall performance as shown in Table 4.

Table 4
Results of recovered records (global message: RECORD) of corrupted files by tools.

Tools	File_1	File_2	File_3	File_4	File_5
Our methods	**3396**	**4889**	**4034**	**5456**	**8546**
GOTOES utility for Strava	1082	3145	583	2864	6290
FIT Viewer Tool by RUNALYZE	1082	3145	205	2864	7537
(FIT SDK) Activity file repair tool	1082	3145	205	2864	6290
FIT File Tools	1082	3145	205	2864	6290
Garmin Fit file conversion and repair	2417	4881	4034	5456	2643

In the recovery log of the File_5, as shown in Fig. 14, our methods first identified the invalid record header and generated regex based on the definition messages. Subsequently, sliding window pattern match method was used to recover additional data.

The comparison through the proof-of-concept and real-world experiments certainly proves our methodology's effectiveness in improving the recovery rate of the file.

7. Conclusion

This study presents a novel and efficient method for in-depth and dense recovery of data from corrupted FIT files. By combining 3 phases of data searching techniques with a thorough understanding of the FIT file structure, we achieved notable success rates in recovering ride data. Our approach provides a valuable contribution to the growing demand for recovering data from bike computers.

Expanding our approach to handle additional global messages in FIT files can enhance the scope of our analysis. The global message 'Lap', for example, represent laps or intervals within the session. It records various data at a specific distance, including start time, start and end position,

---

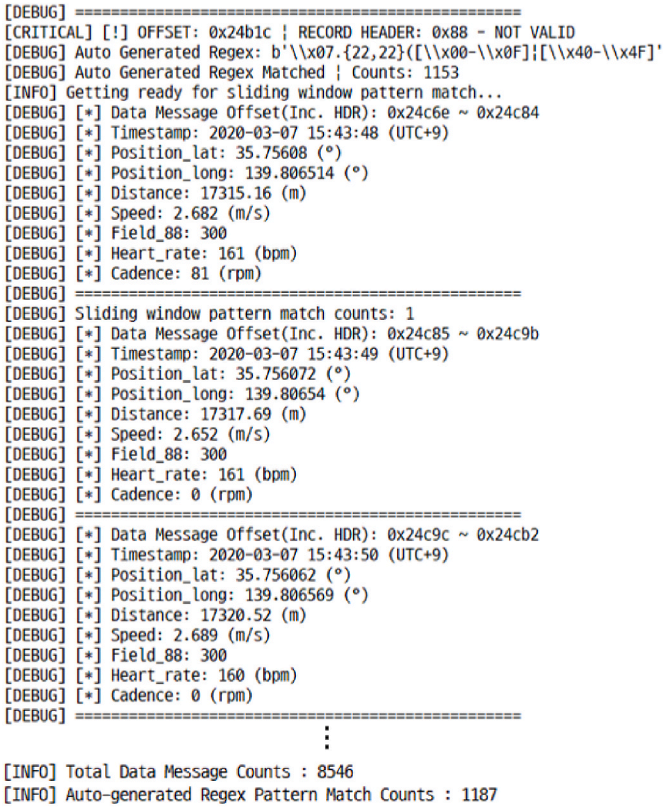
[1] https://github.com/kyl3song/BikeComputer_FIT_file_Recovery.

[2] File1,2: https://forums.garmin.com/apps-software/mobile-apps-web/f/garmin-connect-web/270775/help-fixing-a-corrupt-fit-file/1575917

File3: https://forums.garmin.com/sports-fitness/running-multisport/f/forerunner-30-35/145630/corrupted-fit-files-that-won-t-upload-to-garmin-connect-express

File4: https://forums.garmin.com/apps-software/mobile-apps-web/f/garmin-connect-web/305001/corrupt-fit-file-repair

File5: https://forums.garmin.com/apps-software/mobile-apps-web/f/garmin-connect-web/217612/corrupt-fit-file-still-possible-to-fix.



```
[DEBUG] ==================================================
[CRITICAL] [!] OFFSET: 0x24b1c ¦ RECORD HEADER: 0x88 - NOT VALID
[DEBUG] Auto Generated Regex: b'\\x07.{22,22}([\\x00-\\x0F]¦[\\x40-\\x4F]'
[DEBUG] Auto Generated Regex Matched ¦ Counts: 1153
[INFO] Getting ready for sliding window pattern match...
[DEBUG] [*] Data Message Offset(Inc. HDR): 0x24c6e ~ 0x24c84
[DEBUG] [*] Timestamp: 2020-03-07 15:43:48 (UTC+9)
[DEBUG] [*] Position_lat: 35.75608 (°)
[DEBUG] [*] Position_long: 139.806514 (°)
[DEBUG] [*] Distance: 17315.16 (m)
[DEBUG] [*] Speed: 2.682 (m/s)
[DEBUG] [*] Field_88: 300
[DEBUG] [*] Heart_rate: 161 (bpm)
[DEBUG] [*] Cadence: 81 (rpm)
[DEBUG] ==================================================
[DEBUG] Sliding window pattern match counts: 1
[DEBUG] [*] Data Message Offset(Inc. HDR): 0x24c85 ~ 0x24c9b
[DEBUG] [*] Timestamp: 2020-03-07 15:43:49 (UTC+9)
[DEBUG] [*] Position_lat: 35.756072 (°)
[DEBUG] [*] Position_long: 139.80654 (°)
[DEBUG] [*] Distance: 17317.69 (m)
[DEBUG] [*] Speed: 2.652 (m/s)
[DEBUG] [*] Field_88: 300
[DEBUG] [*] Heart_rate: 161 (bpm)
[DEBUG] [*] Cadence: 0 (rpm)
[DEBUG] ==================================================
[DEBUG] [*] Data Message Offset(Inc. HDR): 0x24c9c ~ 0x24cb2
[DEBUG] [*] Timestamp: 2020-03-07 15:43:50 (UTC+9)
[DEBUG] [*] Position_lat: 35.756062 (°)
[DEBUG] [*] Position_long: 139.806569 (°)
[DEBUG] [*] Distance: 17320.52 (m)
[DEBUG] [*] Speed: 2.689 (m/s)
[DEBUG] [*] Field_88: 300
[DEBUG] [*] Heart_rate: 160 (bpm)
[DEBUG] [*] Cadence: 0 (rpm)
[DEBUG] ==================================================
                                ⋮
[INFO] Total Data Message Counts : 8546
[INFO] Auto-generated Regex Pattern Match Counts : 1187
```

Fig. 14. Recovery log of the File_5 by our tool.

total elapsed time, and more. These data provide valuable insights that can be informative in further analyzing user activities.

Furthermore, our analysis extends to include other fitness tracking devices, such as the Garmin Forerunner, which also utilize the FIT format. By integrating these additional data sources into our current approach, we can enhance the scope and depth of our study in the field of fitness device forensics.

References

Custom Market Insights, 2023. Global GPS bike computer Market 2023–2032. https://www.custommarketinsights.com/report/gps-bike-computer-market/.

Dawson, L., Akinbi, A., 2021. Challenges and opportunities for wearable IoT forensics: TomTom Spark 3 as a case study. Forensic Sci. Int.: Report 3. https://doi.org/10.1016/j.fsir.2021.100198.

Garmin LTD or Its Subsidiaries. Fit SDK. https://developer.garmin.com/fit/download/. (Accessed 13 May 2023).

Garmin Canada Inc. ANT+ advantages. https://www.thisisant.com/business/why-ant/ant-advantages/. (Accessed 13 May 2023).

Garmin Canada Inc. Ant/ANT+ defined. https://www.thisisant.com/developer/ant-plus/ant-antplus-defined. (Accessed 13 May 2023).

Garmin Forums, 2020. Corrupt FIT file. https://forums.garmin.com/apps-software/mobile-apps-web/f/garmin-connect-web/228040/corrupt-fit-file.

Garmin LTD or Its Subsidiaries, 2023. Decoding FIT Activity Files. https://developer.garmin.com/fit/cookbook/decoding-activity-files/. (Accessed 13 May 2023).

Hantke, F., Dewald, A., 2020. How can data from fitness trackers be obtained and analyzed with a forensic approach? Proceedings - 5th IEEE European Symposium on Security and Privacy Workshops, Euro S and PW 2020, 500–508. https://doi.org/10.1109/EuroSPW51379.2020.00073.

Hossein, Fereidooni, Frassetto, Tommaso, Miettinen, Markus, Sadeghi, Ahmad-Reza, Conti, Mauro, 2017. Fitness trackers: fit for health but unfit for security and privacy. In: 2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE), pp. 19–24. https://doi.org/10.1109/chase.2017.20.

Hutchinson, S., Mirza, M.M., West, N., Karabiyik, U., Rogers, M.K., Mukherjee, T., Aggarwal, S., Chung, H., Pettus-Davis, C., 2022. Investigating wearable fitness applications: data privacy and digital forensics analysis on android. Appl. Sci. 12 (19) https://doi.org/10.3390/app12199747.

Kang, S., Kim, S., Kim, J., 2020. Forensic analysis for IoT fitness trackers and its application. Peer-to-Peer Networking and Applications 13 (2), 564–573. https://doi.org/10.1007/s12083-018-0708-3.

Kim, M., Shin, Y., Jo, W., Shon, T., 2023. Digital forensic analysis of intelligent and smart IoT devices. J. Supercomput. 79 (1), 973–997. https://doi.org/10.1007/s11227-022-04639-5.

Mordor Intelligence, 2023. Gps bike computers market size & share analysis - growth trends & forecasts (2023 - 2028). https://www.mordorintelligence.com/industry-reports/gps-bike-computers-market.

Strava. Running, cycling & hiking app - train, track & share. https://www.strava.com/. (Accessed 13 May 2023).

Strava, 2022. Uploading poorly formatted GPS files. https://support.strava.com/hc/en-us/articles/216917967-Uploading-Poorly-Formatted-GPS-Files.

Garmin LTD or Its Subsidiaries, 2023. Flexible and Interoperable Data Transfer (FIT) Protocol. https://developer.garmin.com/fit/protocol/. (Accessed 13 May 2023).

MacDermott, Á., Lea, S., Iqbal, F., Idowu, I., Shah, B.. Forensic analysis of wearable devices: Fitbit, Garmin and HETP watches. https://doi.org/10.1109/NTMS.2019.8763834.