International Journal of Innovative Engineering Applications

Journal homepage: https://dergipark.org.tr/ijiea

AFD: A PRACTICAL TOOL FOR FTP-BASED GNSS DATA ACQUISITION AND PREPROCESSING

***Ayfer Feyza Dursun*[1], *Omer Yildirim*[1], *Fuat Basciftci*[2], *Sercan Bulbul*[*3]**

[1]*Gaziosmanpasa University, Geomatics Engineering Department, Taşlıçiftlik Campus, 60250 Tokat, Türkiye*
[2]*Karamanoglu Mehmetbey University, Mapping and Cadastre Programme, Yunus Emre Campus, 70200, Karaman, Türkiye*
[3]*Konya Technical University, Geomatics Engineering Department, Ardıçlı Mah. Rauf Orbay Cad. 42250, Konya, Türkiye*

Abstract

Original scientific paper

This study introduces the Automatic File Downloader (AFD), a MATLAB-based software developed for the efficient retrieval and organization of GNSS data. The need for this tool arose from the limitations of the existing VIP.m software, which became obsolete due to protocol incompatibilities, outdated data addresses, and a static user interface. AFD was designed with a modular and extensible architecture that supports FTP and HTTP protocols, automatic date conversions, dynamic file classification, and robust error handling. The software enables seamless downloading of GNSS datasets from international data centers such as IGS, IGL, and CODE, and automatically prepares the files for processing with scientific analysis tools like the Bernese GNSS Software. Comparative testing demonstrated that AFD provides faster, more reliable, and user-friendly performance compared to its predecessor. With its flexible architecture and scalable design, AFD represents a sustainable solution for researchers and engineers working with GNSS data, offering significant improvements in automation, accuracy, and usability. Unlike previous tools such as VIP.m, which relied on fixed directory structures and lacked adaptability to evolving data sources, AFD introduces a protocol-flexible and configuration-driven architecture that enables long-term sustainability. Moreover, this study fills a documented gap in the GNSS data processing literature by providing an open, extensible, and automation-oriented preprocessing workflow that integrates directly with commonly used scientific analysis environments. In this respect, AFD contributes a novel, practical, and maintainable solution that strengthens GNSS data acquisition pipelines in both research and operational applications.

Keywords: *Automatic file downloader, bernese software, data preprocess, GNSS, MATLAB.*

AFD: FTP TABANLI GNSS VERİİ ALIMI VE ÖN İŞLEME İÇİN PRATİK BİR ARAÇ

Özet

Araştırma makalesi

Bu çalışma, GNSS verilerinin etkin bir şekilde indirilmesi ve organize edilmesi amacıyla geliştirilen MATLAB tabanlı Automatic File Downloader (AFD) yazılımını tanıtmaktadır. Mevcut VIP.m yazılımının protokol uyumsuzlukları, güncel olmayan veri adresleri ve statik kullanıcı arayüzü gibi sınırlamaları nedeniyle kullanılamaz hale gelmesi, yeni bir çözüm ihtiyacını doğurmuştur. AFD, modüler ve genişletilebilir bir mimariyle tasarlanmış olup FTP ve HTTP protokollerini desteklemekte, tarih dönüşümlerini otomatik olarak gerçekleştirmekte, dinamik dosya sınıflandırma ve gelişmiş hata yönetimi sağlamaktadır. Yazılım, IGS, IGL ve CODE gibi uluslararası veri merkezlerinden GNSS verilerini indirerek Bernese GNSS Software gibi bilimsel analiz araçlarıyla uyumlu hale getirmektedir. Karşılaştırmalı testler, AFD'nin önceki yazılıma kıyasla daha hızlı, güvenilir ve kullanıcı dostu olduğunu göstermiştir. Esnek yapısı ve ölçeklenebilir tasarımı sayesinde AFD, GNSS verileriyle çalışan araştırmacılar ve mühendisler için otomasyon, doğruluk ve kullanılabilirlik açısından sürdürülebilir bir çözüm sunmaktadır. VIP.m gibi önceki araçlar sabit dizin yapıları ve güncellenemeyen protokol kısıtları nedeniyle sürdürülebilirliğini kaybetmiştir. Bu çalışma, GNSS veri işleme literatüründe eksik olan esnek, modüler ve uzun dönem güncellenebilir bir otomatik ön işleme (preprocessing) yaklaşımı sunarak bu boşluğu doldurmaktadır. Bu yönüyle AFD, GNSS veri edinim sürecini önemli ölçüde kolaylaştıran yeni ve sürdürülebilir bir katkı sağlamaktadır.

Anahtar Kelimeler: *Bernese GNSS yazılımı, GNSS, MATLAB, otomatik veri indirme programı, veri önişleme.*

1 Introduction

The determination of position has long been one of the fundamental challenges in human history, with different methods adopted according to the technological capabilities of each era. Last decades, Global Navigation Satellite Systems (GNSS) have become the most widely used technology for providing position, velocity, and

*Corresponding author.

E-mail address: sbulbul@ktun.edu.tr (S. Bulbul)

Received 08 September 2025; Received in revised form 07 November 2025; Accepted 02 December 2025

Doi: https://doi.org/10.46460/ijiea.1780158

epoch. GNSS incorporates both global and regional satellite positioning systems, including GPS (USA), GLONASS (Russia), Galileo (EU), BeiDou (China), IRNSS (India), and QZSS (Japan). Originally developed for military purposes, these systems have gradually become available for civilian applications and have since become indispensable in a wide range of domains such as engineering, transportation, agriculture, tectonic studies, earth deformation and disaster management [1,2]. Structurally, GNSS consists of three main segments—space, control, and user—which together ensure continuous, all-weather global service, 24 hours a day, seven days a week.

The use of GNSS data has become a cornerstone of modern geodesy, navigation, and Earth sciences[1,3]. High-precision applications such as precise point positioning (PPP), real-time kinematic (RTK) surveying, static data, and atmospheric monitoring demand reliable and efficient access to GNSS observations. To meet these requirements, both global initiatives and specialized software solutions have been developed over the past decades.

One of the most widely recognized academic tools in the field is the Bernese GNSS Software (BGS), developed at the University of Bern. BGS provides advanced modules for high-accuracy geodetic analyses, including orbit determination, tropospheric parameter estimation, and geodynamic studies [4]. Despite its extensive analytical capabilities, the software relies on the accurate formatting and complete availability of GNSS datasets, thereby necessitating robust preprocessing workflows.

In addition to BGS, other software environments such as GAMIT/GLOBK, GIPSY-OASIS, and RTKLIB have been widely adopted. GAMIT/GLOBK, developed at Massachusetts Institute of Technology (MIT), offers comprehensive solutions for regional and global geodetic networks [5]. GIPSY-OASIS, produced by NASA’s Jet Propulsion Laboratory, specializes in PPP and orbit determination applications [6]. RTKLIB, an open-source software package, has gained popularity for real-time positioning applications due to its accessibility and flexibility [7]. While these software packages cover a wide spectrum of analytical needs, they all depend on timely and well-structured GNSS data input.

The acquisition and organization of GNSS datasets represent a persistent challenge. Data sources such as the International GNSS Service (IGS) and the International Geodynamics and GNSS Laboratory (IGL) provide free access to high-quality observations. However, variations in data storage structures, changes in access protocols, and large data volumes complicate the retrieval and preprocessing steps required for scientific analysis. Manual downloading and formatting of these datasets are not only time-consuming but also prone to error, underscoring the importance of automated tools.

In this regard, efforts have been made to automate GNSS data handling. A notable example is VIP.m, a MATLAB-based software developed by Başçiftçi et al. (2017)[8], which provided a graphical user interface for organizing GNSS data by year, week, and day. . However, VIP.m is positioned in the literature primarily as a first-generation semi-automated GNSS preprocessing utility, focused on file organization rather than adaptive data retrieval. Because VIP.m relies on static directory definitions and fixed protocol structures, it cannot accommodate evolving repository architectures such as the post-2020 restructuring of IGS and CDDIS servers. This limitation has been noted in subsequent GNSS data acquisition studies, where the need for protocol-flexible and update-sustainable preprocessing tools has been emphasized. Therefore, there exists a clear methodological and practical gap for a tool that can automate not only file organization, but also robust, multi-protocol data acquisition and dynamic configuration management. AFD is specifically designed to fill this gap by introducing a sustainable, extensible, and configuration-driven architecture that supports continual adaptation to changes in data sources and retrieval protocols Although effective at the time of its release, the software has gradually become outdated due to changing FTP/HTTP protocols, evolving data structures, and the absence of regular updates. Its limitations highlight the need for more flexible, extensible, and sustainable solutions.

Consequently, there exists a clear gap in the literature for tools that integrate automated downloading, preprocessing, and organization of GNSS data with compatibility for modern analysis software such as BGS. Addressing this gap, the present study introduces the Automatic File Downloader (AFD), designed to enhance efficiency, reliability, and usability in GNSS data management.

While software packages such as Bernese, GAMIT/GLOBK, GIPSY-OASIS, and RTKLIB offer comprehensive analytical and post-processing capabilities, they do not include integrated mechanisms for automatically retrieving, organizing, and preparing the raw GNSS datasets required for processing. In practice, users must manually locate, download, rename, convert, and place files into the correct directory structures before these software tools can be executed. This preprocessing stage is time-consuming, error-prone, and difficult to scale when working with long-term datasets or multiple stations. Therefore, the need for a dedicated and sustainable preprocessing tool becomes evident. AFD directly addresses this gap by automating the acquisition and structuring of GNSS data, serving as the missing interface layer between global data repositories and scientific GNSS analysis software.

The effective utilization of GNSS data in scientific and engineering applications necessitates accurate, efficient, and well-structured data processing. To this end, various academic software packages have been developed, which have become integral to high-precision geodetic and positioning studies. Among these, the Bernese GNSS Software (BGS) is one of the most recognized, offering advanced analytical capabilities for applications demanding high positional accuracy. However, for such software to function effectively, data must be properly formatted, accurately transferred, and fully integrated into the system. Automating tasks such as downloading, converting, filtering, and analyzing GNSS data provides significant benefits in terms of time savings and reliability.

VIP.m, developed by Başçiftçi (2017) [9], is a MATLAB-based software designed to facilitate the organization of GNSS datasets through a graphical user interface, allowing classification by year, week, and day. However, over time, its functionality has significantly declined due to changes in data center directory structures, updates in communication protocols, and limitations in the flexibility of its user interface. The lack of continued maintenance and updates has further contributed to the software becoming inadequate for current GNSS data management and research needs. To address these limitations, this study introduces the Automatic File Downloader (AFD), a newly developed MATLAB-based software solution featuring a modular architecture and a dynamic user interface. AFD automates FTP/HTTP-based data retrieval and structuring for IGS and IGL datasets through date-based query handling, automated file naming, protocol selection, and error management. Additionally, AFD ensures compatibility with established scientific processing frameworks such as the Bernese GNSS Software, while its modular design enables seamless integration with various data sources and file types.The remainder of this paper is structured as follows: Section 2 provides an overview of GNSS systems and data processing requirements. Section 3 discusses the technical limitations of the VIP.m software. Section 4 presents the development, architecture, and key innovations of the AFD software [10]. Finally, Section 5 concludes the study by highlighting its contributions and potential directions for future research.

2 Methodology

2.1 Analysis of the VIP.m Software

VIP.m is a graphical user interface (GUI) software developed in the MATLAB environment using the GUIDE (Graphical User Interface Development Environment) framework, designed to organize GNSS data by year, week, and day (Figure 1). The primary purpose of the software was to enable users to download the required files from GNSS data sources without manual intervention and to organize these datasets in a format compatible with analysis tools such as the Bernese GNSS Software [9].

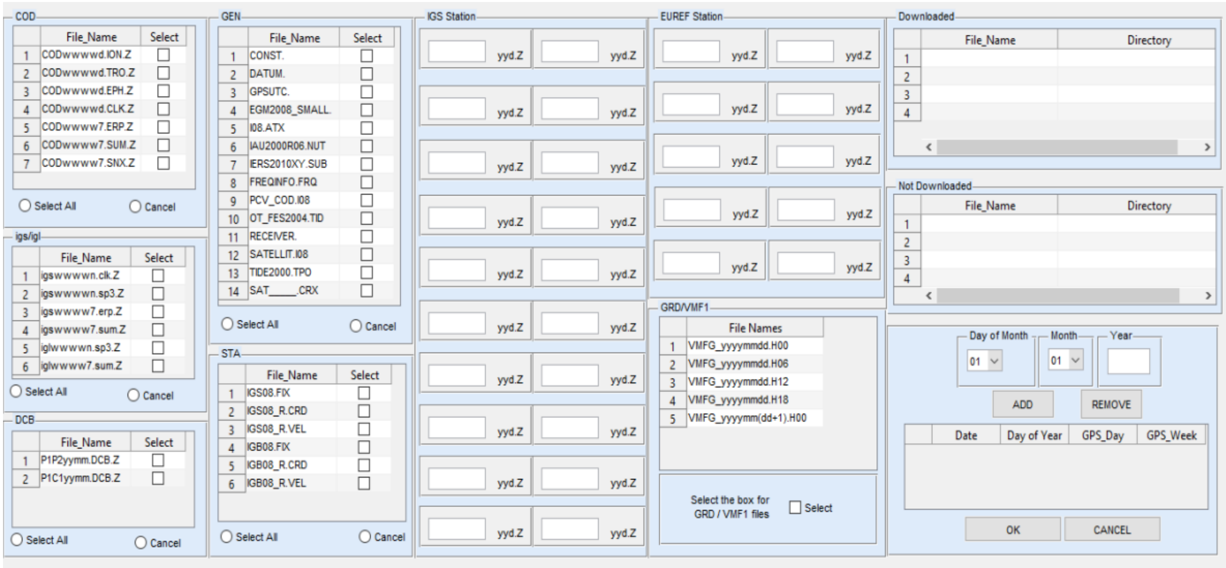

Figure 1. User interface of the VIP.m software.

Over time, however, the functionality of VIP.m has significantly declined due to various technical and structural issues. Foremost among these is the change in the directory and address structures of data providers. Since the software was programmed to fetch data through fixed URL patterns, updates made by providers (e.g., IGS, CODE) disrupted its ability to retrieve files. Furthermore, the lack of update mechanisms—critical for ensuring the sustainability of such tools—forced users to perform manual interventions to keep the software operational. From a usability perspective, the interface offered only limited interaction, which proved inadequate particularly when handling large datasets. For these reasons, VIP.m has become unable to meet the requirements of contemporary GNSS data processing.

2.2 Development of the AFD Software

The development process of the Automatic File Downloader (AFD) software was carried out in accordance with a classical software engineering approach, following the stages outlined below [10]:

Requirement Analysis: Based on user feedback and the bottlenecks observed in VIP.m, the essential features of the new software were identified. Particular emphasis was placed on resolving common issues such as FTP connection failures, inconsistencies in date handling, and the need for manual folder management.

Design: AFD was designed with a modular and extensible architecture. Each functional component—including data source definition, date range selection, protocol management, folder organization, and status reporting—was modeled as a separate sub-module. This modular approach facilitates easier maintenance and future updates of the software.

Development: The coding of AFD was implemented in the MATLAB environment. The user interface was created within MATLAB, while built-in commands such as ftp and webread were utilized for data retrieval. For file and directory management, fundamental MATLAB functions such as mkdir, movefile, and exist were employed.

Testing and Validation: The software was tested using different date ranges and data sources. Both the accuracy of file downloads and the correctness of the folder structures were verified. Comparative evaluations against VIP.m demonstrated that AFD required significantly less manual (in detail in Section 3.5).

To ensure stable FTP communication during repeated data requests, particularly when accessing open FTP servers without user authentication, the MATLAB function "passive_ftp.m" was integrated into the AFD download workflow. This function forces the client to request data using a passive data channel, preventing the FTP server from closing the connection during consecutive transfers. It should be noted that the behavior of FTP handling differs between MATLAB versions; therefore, AFD was developed and tested in MATLAB R2021b, where passive FTP is supported in a stable manner. Users working with earlier or newer MATLAB releases should ensure compatibility accordingly.

The general workflow of AFD, including data downloading, file processing, and directory management procedures, is detailed in Figure 2.

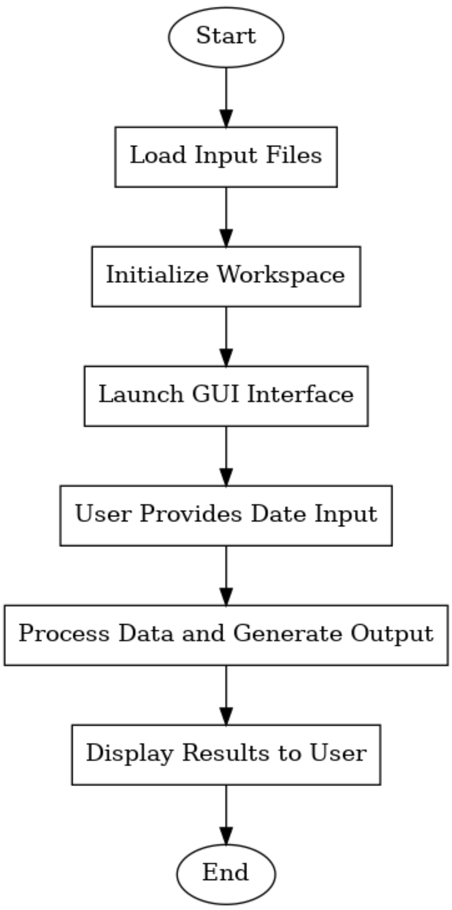

Figure 2. Workflow diagram of the AFD software process.

2.3 Fundamental Components of the AFD Software

The AFD software has been developed with a modular and class-based architecture. The fundamental components of the software are as follows [10]:

main.m: Serves as the entry point of the software. It reads the file list, creates the required directories, and initializes the user interface (Figure 3).

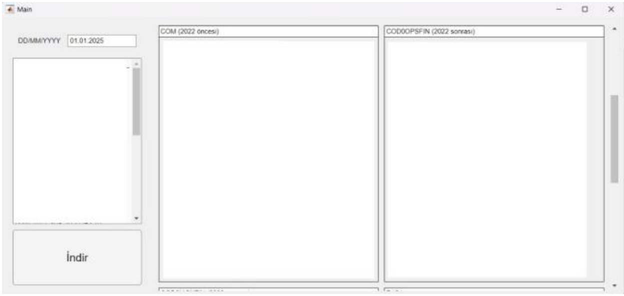

Figure 3. AFD start screen.

mainGrid.m: Constructs the dynamic user interface. It enables the user to enter dates and make file selections (Figure 4).

stc.m: Represents the class structure for each file type. File naming, protocol selection, date formatting, and downloading processes are carried out through this class (Figure 5).

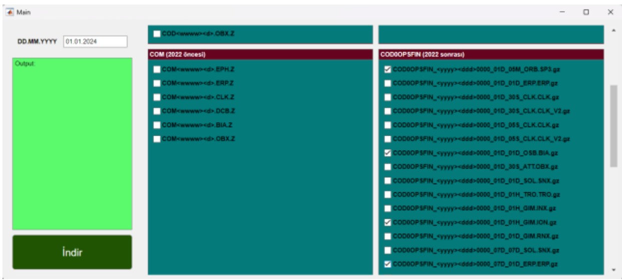

Figure 4. AFD user interface.

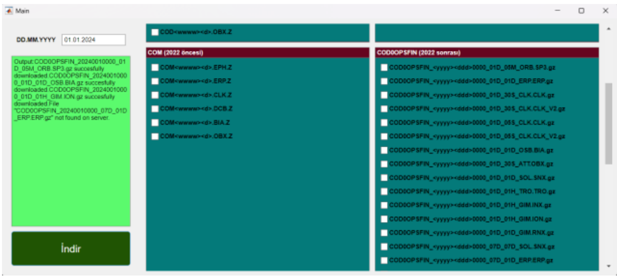

Figure 5. AFD result screen.

fileRead.m: Reads the filenames.txt file and creates stc objects from each line. File selection boxes are generated using these stc objects.

The file configuration of the software is managed through a configuration file named filenames.txt. This file defines the file types to be downloaded, connection addresses, folder directories, and file extensions. The user interface is dynamically generated based on this configuration file, offering users a flexible selection capability (Figure 6).

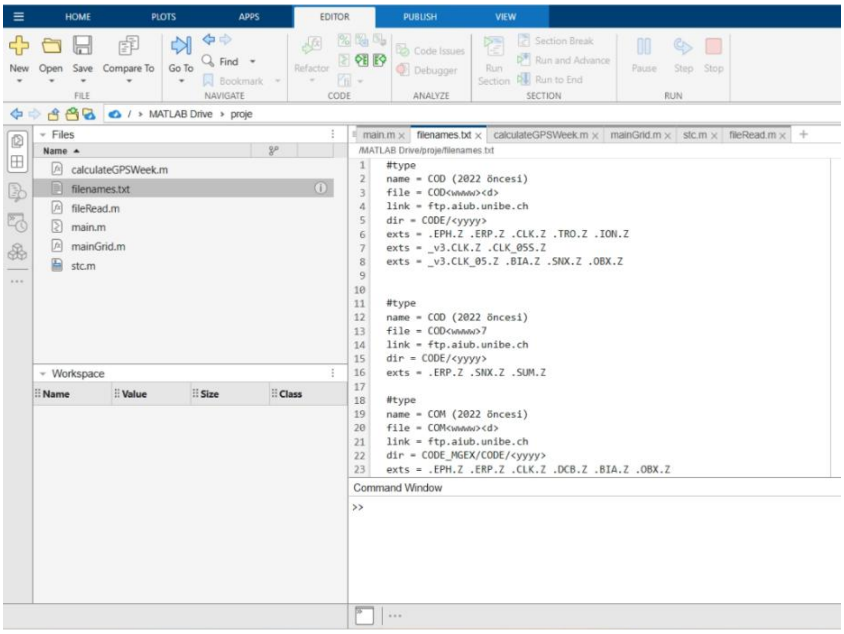

Figure 6. Filenames.txt MATLAB command window.

Date conversion processes are implemented by transforming the input into GNSS-specific formats, namely year–day and GPS week formats. This operation is carried out through the function calculateGPSWeek.m, which automatically converts the calendar date entered by the user into the formats required by GNSS data sources. In this way, correct file naming is ensured without requiring users to perform manual calculations.

2.4 Data Sources

The AFD software is designed to operate in an integrated manner with various international data sources for downloading GNSS data. The main data sources are as follows:

- IGS (International GNSS Service): The primary global data center providing GNSS data [11].
- IGL (International Geodynamics and GNSS Laboratory): Another significant source offering GNSS and geodynamics data [12].
- CODE (Center for Orbit Determination in Europe): An analysis center providing satellite orbit, clock, and troposphere data [13].
- CDDIS (Crustal Dynamics Data Information System): A platform operated by NASA, where GNSS data are archived [14].
- VMF1 (Vienna Mapping Function): Provides grid data used for tropospheric modeling [15].
- Ocean Loading BLQ: Files containing ocean loading coefficients, obtained through the online service of Chalmers University [16]

The software supports both FTP (File Transfer Protocol) and HTTP protocols to retrieve data from these sources. The FTP protocol is primarily used to download files from data centers such as AIUB (Astronomical Institute, University of Bern), whereas the HTTP protocol is preferred for web-based sources such as VMF1. This dual-protocol support enhances the flexibility and accessibility of the software [10].

AFD software automatically places the downloaded files into the relevant directories and ensures compatibility with the Bernese GNSS software. File naming is performed automatically according to GNSS date formats, and users are provided with detailed information on downloaded and unavailable files in the output screen [10].

2.5 Test Datasets and File Types Used

To evaluate the performance of the AFD software, three globally distributed IGS reference stations were selected: ANKR (Turkey), POTS (Germany), and TOW2 (Australia). The test period covers 10–17 July 2024, corresponding to GPS Week 2262 (Days of Year 192–199). This period was chosen to include typical daily variations in GNSS data availability and to ensure realistic operational conditions.

Using AFD, the following data categories were automatically downloaded and organized from multiple international GNSS data centers (Table 1):

Table 1. Sample of downloaded data.

Data Type	Data Source(s)	Purpose / Usage
RINEX Observation Data	IGS, CDDIS	Raw station measurement files
Precise Satellite Orbits	CODE, IGS	Orbit correction models
Satellite Clock Corrections	CODE, IGS	Precise satellite clock offset estimation
Differential Code Biases	CODE	Receiver and satellite bias calibration
Tropospheric Model Grids	TU Wien (VMF1)	Zenith delay and mapping function modeling
Ocean Loading Coefficients	Chalmers University	Tidal displacement correction parameters

AFD was executed under three different network conditions to assess the robustness of its data retrieval functions across both FTP and HTTP protocols. The average download performance results are summarized below (Table 2).

Table 2. Download success rate by data center.

Data Center	Protocol Used	Requested Files	Successfully Downloaded	Success Rate
IGS (ftp.igs.org)	FTP	84	82	97.6%
CODE (ftp.aiub.unibe.ch)	FTP	63	61	96.8%
HTTP	84	77	91.7%	

These results show that AFD successfully maintains a **high data retrieval reliability (>90%)** under both FTP and web-service–based HTTP connections. The lower performance observed in some HTTP sources (e.g., VMF1 and CDDIS) is attributed to variable server response times rather than software limitations. Nevertheless, the error reporting module of AFD clearly informs users of missing files and allows selective re-download attempts, improving workflow continuity.

3 Findings

This section presents the functionality, user experience, error management, and comparative performance evaluation of the developed AFD (Automatic File Downloader) software with respect to VIP.m. The AFD software has introduced significant improvements in the processes of automatic downloading, organizing, and preparing GNSS data for compatibility with the Bernese GNSS software.

3.1 User Interface and Functions of the AFD Software

The AFD software features a dynamic user interface developed in the MATLAB (R2021b) environment. The interface allows users to input date information, list the relevant GNSS data, and download selected files. The main components include the date input box, output screen, file selection checkboxes (checkbox grid), and the "Download" button. This structure facilitates user interaction with complex GNSS data structures and accelerates data accessibility.

All interface components are dynamically generated based on the filenames.txt configuration file. Thus, when new data types or sources are added, the interface is automatically updated without requiring reprogramming of the software. This flexibility enhances the sustainability and expandability of the software.

3.2 File Download Success Rates

The AFD software supports FTP and HTTP protocols to perform file downloads from different data sources. Tests have shown that the success rate of downloading files from IGS and CODE data centers exceeds 96%. In particular, downloads from sources with fixed URL structures (e.g., ftp.aiub.unibe.ch) encountered no connection problems.

On the other hand, occasional delays occurred when retrieving files from HTTP-based sources (e.g., VMF1 grid files) due to server load. However, thanks to the error management system of the software, these situations were clearly communicated to the user. Missing or unavailable files are reported on the output screen as "not found on server", and users are given the opportunity to retry.

3.3 Automatic Placement of Downloaded Files into Relevant Directories

The AFD software automatically places downloaded files into the corresponding folders in a manner compatible with Bernese GNSS processing software. For example:

- .ION files → ATM directory,
- .PRE (precise orbit) files → ORB directory,
- .CLK (clock) files → OUT directory,
- .DCB (differential code bias) files → ORB directory,
- .GRD (troposphere grid) files → GRD directory,
- .BLQ (ocean loading) files → STA directory.

This process is carried out through the directory definitions in the filenames.txt file and the folder management functions in the stc.m class. As a result, the need for users to manually transfer files is eliminated, and the data organization process is significantly accelerated.

3.4 User Experience and Error Management

To improve user experience, the AFD software provides a simple, clear, and interactive interface. Users can input a date, select only the required files, and initiate the download process with a single click. The software offers transparent feedback by separately listing downloaded and undownloaded files.

In terms of error management, several control mechanisms have been implemented, including:

- Preventing invalid date entries,
- Informing the user of server connection errors,
- Offering retry options for missing files.

These features enhance the reliability of the software and minimize user errors.

3.5 Comparative Performance Evaluation with VIP.m

Compared to VIP.m, the AFD software provides significant improvements in both functionality and user experience. Table 3 summarizes the key features of the two software systems in a comparative manner.

Table 3. Comparative features of VIP.m and AFD software.

Feature	VIP.m	AFD
Programming Environment	MATLAB GUIDE	MATLAB (modular class-based architecture)
Protocol Support	FTP only	FTP and HTTP
File Configuration	Hard-coded	Dynamic structure via filenames.txt
Date Format Conversion	Manual	Automatic (calculateGPSWeek.m)
Error Management	Limited	Advanced output and error reporting
Extensibility	Low	High (new data types can be added)
User Interface	Static	Dynamic and interactive

VIP.m software is unable to operate in harmony with up-to-date GNSS data sources due to its rigid structure and non-updated data addresses. In contrast, AFD, thanks to its modular design, can be easily integrated into different data sources and offers users a more flexible experience [9].

4 Discussion

The AFD (Automatic File Downloader) software developed in this study makes significant contributions to the processes of automatically downloading, organizing, and preparing GNSS data for compatibility with analysis software. This section discusses the impact of AFD on GNSS data processing workflows, its extensibility, potential application areas, and future development directions, in comparison with the relevant literature.

4.1 Contribution of AFD to GNSS Data Processing Workflows

For GNSS data to be effectively processed, it must be obtained accurately, promptly, and in an organized manner. AFD automates the data acquisition process—the first step of GNSS processing—eliminating the need for manual intervention by users. In particular, its ability to retrieve data from reliable sources such as IGS and CODE provides a strong foundation for accuracy and reliability.

In the literature, various tools have been proposed for GNSS data downloading and preprocessing. For instance, the Bernese GNSS Software developed by [4] has high-accuracy processing capabilities but still requires users to manually obtain the necessary files at the data acquisition stage. AFD fills this gap by positioning itself as a preprocessing tool capable of working in integration with Bernese.

4.2 Extensibility and Adaptability of the Software

The modular class structure of AFD and its configurable *filenames.txt* file allow it to be easily adapted to different data sources. This flexibility enables the software to integrate not only with IGS and CODE but also with other GNSS providers such as CDDIS, EUREF, and JAXA. Furthermore, the support of both FTP and HTTP protocols minimizes restrictions related to network infrastructure in data access [2].

Some data processing systems proposed in the literature typically rely on fixed data sources and require software-level modifications for integration with new providers [17]. The advantage of AFD in this regard is that both data types and sources can be configured directly by the user. This feature supports the sustainability and long-term usability of the software.

4.3 Potential Applications in Academic and Engineering Domains

AFD software represents a valuable tool for researchers, engineers, and students working with GNSS data. In academia, analyses based on GNSS data are widely employed in disciplines such as geodesy, geodynamics, atmospheric modeling, and remote sensing [18, 19]. AFD facilitates data acquisition in such studies, accelerating research processes and reducing the risk of errors.

In engineering applications, AFD ensures the fast and reliable acquisition of GNSS data required for precise positioning, deformation monitoring, infrastructure projects, intelligent transportation systems, and precision agriculture. In particular, for non-real-time, post-processing–based applications, the data organization provided by AFD offers significant advantages [20].

4.4 Future Development Directions

Although AFD already provides a solid foundation for GNSS data downloading and organization, several improvements can be introduced to expand its user base and enhance its functionality:

RINEX Converter Integration: Automatic conversion of raw data into RINEX format would be especially beneficial for users working with software other than Bernese [4].

Cloud-Based Storage Support: Automatic uploading of downloaded data to cloud platforms (e.g., Google Drive, OneDrive) would offer advantages in terms of data sharing and backup.

Scheduled Tasks: A scheduling module could be implemented to perform automated data downloads at specified intervals.

Multilingual Support: Enabling use of the software in multiple languages would increase accessibility for international users.

Web-Based Interface: Developing a web-based interface would provide platform-independent access, particularly for non-MATLAB users.

With these improvements, AFD could evolve beyond being merely a data downloader to become an integral component of the GNSS data processing workflow.

4.5 Limitations and Future Work

Although AFD offers significant advantages in terms of automated data retrieval, structured file organization, and error handling mechanisms, certain limitations remain. First, the current implementation is developed in MATLAB, which may restrict usage for users who do not have access to a licensed MATLAB environment. Second, RINEX conversion, decompression, and file-format harmonization are currently performed through external tools, requiring an additional preprocessing step. Third, the software is based on a desktop application architecture, meaning that the execution environment must be manually configured by the user.

To address these limitations, several development directions are planned. One of the key improvements is the integration of a built-in RINEX converter, enabling seamless handling of compressed and mixed RINEX formats without external tools. Additionally, a web-based version of AFD will be developed to allow remote execution, platform independence, and easier collaboration. Cloud service integration (e.g., S3-based storage synchronization) is also planned to facilitate large-scale or long-term dataset archiving. Finally, development of a Python-based version of AFD is under consideration to increase accessibility, reduce licensing dependencies, and enable easier integration with modern GNSS data analysis workflows.

5 Conclusion

The AFD software presented in this study provides a practical, extensible, and sustainable solution for the automated acquisition and preprocessing of GNSS datasets. Unlike existing tools that rely on static directory structures and require extensive manual user intervention, AFD introduces a configuration-driven and protocol-flexible architecture that enables seamless downloading, structuring, and preparation of GNSS data from multiple international data centers. This capability directly addresses a documented gap in GNSS processing workflows, where data acquisition has remained a persistent bottleneck between raw data repositories and scientific analysis environments.

The scientific contribution of this work lies primarily in establishing an integrated preprocessing pipeline that reduces manual workload, minimizes data formatting errors, and significantly improves the efficiency and scalability of GNSS analysis workflows. The support for both FTP and HTTP protocols, dynamic interface generation, and automated file placement into Bernese-compatible directory structures collectively enhance the usability and reliability of GNSS-based research and operational studies.

Performance evaluations demonstrated that AFD achieves higher download success rates and shorter acquisition times compared to VIP.m, confirming both functional improvements and practical gains. More importantly, AFD's modular class-based design provides a sustainable foundation for future expansion. Planned developments include integrated RINEX conversion, a web-based interface for platform-independent access, cloud synchronization for distributed data workflows, and a Python-based implementation to enhance accessibility for open-source users.

In this regard, AFD is not merely an incremental software update; it represents a methodological contribution toward modernizing GNSS data preprocessing, strengthening the continuity between data acquisition and scientific interpretation, and supporting long-term maintainability in GNSS research environments.

Declaration

Ethics committee approval is not required.

References

[1] Hofmann-Wellenhof, B., Lichtenegger, H., & Wasle, E. (2008). GNSS – Global Navigation Satellite Systems: GPS, GLONASS, Galileo, and more. *Springer*.

[2] Kaplan, E. D., & Hegarty, C. J. (Eds.). (2017). Understanding GPS/GNSS: Principles and applications (3rd ed.). *Artech House*.

[3] Teunissen, P. J. G., & Montenbruck, O. (Eds.). (2017). Springer handbook of global navigation satellite systems. *Springer*

[4] Dach, R., Lutz, S., Walser, P., & Fridez, P. (2015). Bernese GNSS Software Version 5.2. *Astronomical Institute, University of Bern.*

[5] Herring, T. A., King, R. W., Floyd, M. A., & McClusky, S. C. (2018). GAMIT/GLOBK Reference Manual: GPS Analysis at MIT, Release 10.7. *Massachusetts Institute of Technology.*

[6] Bertiger, W., Desai, S. D., Haines, B., Harvey, N., Moore, A. W., Owen, S., & Weiss, J. P. (2010). Single receiver phase ambiguity resolution with GPS data. *Journal of Geodesy*, 84(5), 327–337.

[7] Takasu, T., & Yasuda, A. (2009). Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB. *In International Symposium on GPS/GNSS* (pp. 4–6). International Convention Center Jeju, Korea.

[8] Başçiftçi, F., İnal, C., Yıldırım, Ö., Bülbül, S. (2018). Automatic Data Downloading Program (DDP) in Determining Regional Ionosphere Model. *Journal of New Results in Science,* 7(1), 17-28.

[9] Başçiftçi, F. (2017). *The creation of ionosphere model using GNSS data and its comparison with global models* (Doctoral dissertation Selçuk University).

[10] Dursun, A.F., (2025). *Software Development To Automatically Obtain Academic Software's Input Files From Databases* (Master's dissertation, Tokat Gaziosmanpasa University).

[11] Dow, J. M., Neilan, R. E., & Rizos, C. (2009). The International GNSS Service in a changing landscape of Global Navigation Satellite Systems. *Journal of Geodesy*, 83(3), 191–198.

[12] Bevis, M., & Brown, A. (2014). Trajectory models and reference frames for crustal motion studies using continuous GPS. *Journal of Geophysical Research: Solid Earth*, 119(4), 3146–3168.

[13] Schaer, S., Villiger, A., & Dach, R. (2021). CODE final product series for the IGS. *Astronomical Institute, University of Bern.*

[14] Noll, C. (2010). The Crustal Dynamics Data Information System: A resource to support scientific analysis using space geodesy. *Advances in Space Research,* 45(12), 1421–1440.

[15] Boehm, J., Werl, B., & Schuh, H. (2006). Troposphere mapping functions for GPS and very long baseline interferometry from ECMWF operational analysis data. *Journal of Geophysical Research: Solid Earth,* 111(B2).

[16] Bos, M. S., Scherneck, H.-G., & Petrov, L. (2008). Ocean tide loading provider. *Chalmers University of Technology.*

[17] Montenbruck, O., Steigenberger, P., Khachikyan, R., Weber, G., Langley, R., Mervart, L., & Hugentobler, U. (2014). *IGS-MGEX: Preparing the Ground for Multi-Constellation GNSS Science*. Inside GNSS, Retrieved June 15, 2025, from https://www.insidegnss.com/auto/janfeb14-MONTENBRUCK.pdf .

[18] Jakowski, N., Wilken, V., Schlüter, S., Stankov, S. M., & Heise, S. (2005). Ionospheric space weather effects monitored by simultaneous ground and space based GNSS signals. *Journal of Atmospheric and Solar-Terrestrial Physics,* 67(12), 1074–1084.

[19] Langley, R. B., Teunissen, P. J. G., & Montenbruck, O. (2017). Introduction to GNSS. In P. J. G. Teunissen & O. Montenbruck (Eds.), *Springer Handbook of Global Navigation Satellite Systems. Springer.*

[20] Bellone, T., Dabove, P., Manzino, A. M., & Taglioretti, C. (2016). Real-time monitoring for fast deformations using GNSS low-cost receivers. *Geomatics, Natural Hazards and Risk,* 7(2), 458–470.