A method for detecting unauthorized interference in network communications using intelligent traffic analysis*

Vladyslav Bilous[1,*,†], Dmytro Bodnenko[1,†], Oleksandra Lokaziuk[1,2,†], Valerii Kozachok[1,†], and Zoreslava Brzhevska[1,†]

[1] Borys Grinchenko Kyiv Metropolitan University, 18/2 Bulvarno-Kudriavska str., 04053 Kyiv, Ukraine
[2] Institute of Mathematics of NAS of Ukraine, 3, Tereshchenkivska Str., Kyiv, 01024, Ukraine

Abstract

The article is devoted to the study of multimodal approaches of artificial intelligence to automated detection of fake news and verification of media content in the modern digital information environment. The integration of natural language processing, computer vision and video data analysis methods within a single intelligent system is considered. Architectural solutions for collecting and processing data from news aggregators and Telegram channels are analysed, as well as methods for verifying the authenticity of photo and video materials, including the detection of manipulation and deepfake content. An analytical approach to evaluating the effectiveness of multimodal systems using classical machine learning metrics is proposed. Based on the analytical model, the author's software "Vidoglyad" was designed, which offers a range of functions for automated detection of fake news and verification of media content based on artificial intelligence. The Vidoglyad software combines the functionality of OSINT tools and multimodal content analysis methods for working with information flows, verifying the authenticity of sources and content, and obtaining digital evidence of manipulation. The results of the study demonstrate that the combination of heterogeneous modalities significantly increases the accuracy and stability of fake detection systems compared to single-modal approaches.

Keywords

multimodal AI, natural language processing, computer vision, deepfake, media content verification, system analysis, data analysis, coding, fake news

1 Introduction

Today's mechanisms for generating and disseminating news content are focused on current digital social and media platforms. The positive trend of open access to information serves as a manipulative platform for the dissemination of disinformation (manipulative messages, distorted content, fake news, half-truths, pseudoscientific information, etc.). The increased frequency of using technologies that combine text, video, and graphic data to generate fake news significantly complicates expert verification of the authenticity of news content. To solve this verification problem, it is advisable to use Open Source Intelligence (OSINT) tools, which enable rapid analysis of data from open sources for inaccuracy, manipulation, and psychological operations (PSYOP). The combination of OSINT and AI technologies, which are based on Natural Language Processing methods, machine learning algorithms, and computer vision tools, has significant potential for ensuring the automation of accurate, effective, and rapid analysis of media data. Automated OSINT applications for detecting fake news are capable of: analysing text, graphic and video data; identifying stylistic, semantic and contextual signs of disinformation; verifying the reliability of sources of information dissemination in the digital space.

* CPITS 2026: Cybersecurity Providing in Information and Telecommunication Systems, February 28, 2026, Kyiv, Ukraine
* Corresponding author.
† These authors contributed equally.
v.bilous@kubg.edu.ua (V. Bilous); d.bodnenko@kubg.edu.ua (D. Bodnenko); o.lokaziuk@kubg.edu.ua (O. Lokaziuk); v.kozachok@kubg.edu.ua (V. Kozachok); z.brzhevska@kubg.edu.ua (Z. Brzhevska)
0000-0001-6915-433X (V. Bilous); 0000-0001-9303-6587 (D. Bodnenko); 0000-0001-9663-251X (O. Lokaziuk); 0000-0003-0072-2567 (V. Kozachok); 0000-0002-7029-9525 (Z. Brzhevska)

An analysis of scientific publications and the market for existing software for the automated detection of fake news reveals: insufficient coverage of the problem in scientific and practical sources; limited functionality of existing software for detecting current multimodal fake news and manipulation. The relevance of this study is due to the growth of disinformation in the digital media space, the complication of data presentation forms due to the spread of multimodal synthetically generated content, and the lack of OSINT-oriented application software for its detection. The development of OSINT software based on the use of artificial intelligence methods for analysing open sources of information is a timely and practically significant task.

The object of the study is the processes of automated detection of fake news and verification of the authenticity of media content in the digital information environment based on open data sources.

The subject of the study is methods, models, and software tools for analysing multimodal media content using artificial intelligence technologies and OSINT approaches.

The purpose of this study is: to develop and scientifically substantiate the Vidoglyad software as an OSINT tool for the automated detection of fake news and verification of media content based on artificial intelligence methods; to design the software architecture (selection and integration of multimodal data analysis models, implementation of the software product on the .NET platform in C#).

2 Literature overview

The taxonomy of machine learning methods, models, and features of machine and deep learning used in Content-Based Fake News Detection is presented in [1]. Developing this direction, the study [2] provides a thorough analytical review of the use of artificial intelligence tools for automatic detection of fake news, with an emphasis on modern algorithmic approaches.

Further refinement of classification methods is carried out in the study [3], which consider the detection of fake news based on FNID conclusions in two formats: two-class (FNID-FakeNewsNet) and six-class (FNID-LIAR). In a broader context, a general overview of fake news, disinformation, and misinformation in social media is presented in the work [4].

Focusing on deepening architectural solutions, the study [5] proposes a multimodal deep learning model for automatic detection of fake news based on capsule neural networks and the BERT language model. A related approach is highlighted in the work [6], which considers a hybrid deep learning model using FastText and Explainable AI methods.

Along with model-oriented approaches, the studies [7, 8, 9, 10] pay special attention to the practical aspects of implementing OSINT technologies in the context of detecting fake information. Concluding the review of current trends, the study [11] presents ensemble methods for improving the robustness of fake news detection systems, combining transformer architectures, natural language processing methods, and classical machine learning algorithms.

3 Research method

The following methods were used in the study: processing and summarising scientific sources devoted to OSINT tools; researching social networks; analysing resources and approaches to automated detection of fake news and verification of media content; developing software; testing of the author's software product to assess the prospects for its further development; visualization of the obtained data to present the research results.

The research was carried out within the framework of: the scientific theme of the Department of Information and Cyber Security named after Professor Volodymyr Buryachok of Borys Grinchenko Kyiv Metropolitan University "Methods and models of ensuring cyber security of information processing systems and functional security of software and technical complexes of critical infrastructure management", DR No. 0122U200483; the complex scientific theme of the

Faculty of Information Technologies and Mathematics "Mathematical methods and digital technologies in education, science, technology", DR No. 0121U111924. This work was supported by a grant from the Simons Foundation International (SFI-PD-Ukraine-00014586, L.O.V).

4 Main material

4.1 Multimodal approaches to detecting fake news

Text modality. The basic functional component for detecting text news is text content. The analysis of multidimensional semantic representation of text data at the document level is based on contextual transformer language models.

To ensure text analysis, the system performs a series of verification steps. The first step is pre-processing, which includes normalisation, tokenisation, and language identification. Next, named entity recognition (NER) algorithms are applied to identify mentions of people, organisations, and geographical objects. Further implementation involves claim extraction, which analyses text fragments for verifiable factual statements. The next stage involves classifying text data according to reliability and verifying it by correlating it with other modalities and alternative data sources.

This multi-level approach reduces the number of false positives characteristic of simple text classifiers.

Visual modality. The visual component of the system is aimed at analysing images accompanying news materials. The main task is to identify inconsistencies between the content stated in the text and the actual visual context. This is achieved by combining deep computer vision methods [12]. At the initial stage, Perceptual hashing is used to detect the reuse of graphic data. This stage allows us to identify precedents for the use of images that have no direct contextual connection with the event being covered or are chronologically outdated. The next stage is to analyse the presence of local manipulations [13,14] (insertion, retouching, substitution) using convolutional neural networks [15]. To identify conceptual conflicts (inconsistencies in the type of events, time, locations), a semantic comparison of text and graphic data is performed by comparing the embeddings of text content and visual features [16].

Video analysis. Video content that combines spatial and temporal characteristics is the most difficult modality to analyse. The system uses a step-by-step approach, which begins with the extraction of key frames to reduce the computational load. Next, compression and spatiotemporal artefacts characteristic of manipulated videos are analysed.

To detect synthetic videos, features related to inconsistencies in micro-expressions, lip movements, and audio tracks are used. The combination of these indicators allows for increased accuracy in detecting deepfake content, even in cases of high-quality generation.

4.2 Architecture of the Vidoglyad software intelligent system

We offer an intelligent system for automated detection of fake news implemented as a multi-level modular architecture on the .NET platform, which provides scalability, flexibility, and the ability to expand functionality in stages. The architectural solution is based on the principles of loose coupling of components, asynchronous data processing, and clear separation of responsibilities between system levels.

1. Data Ingestion Layer

The data ingestion layer is responsible for asynchronous ingestion of information from heterogeneous sources, including news aggregators, RSS feeds, media websites, and Telegram channels. The ability to process high-intensity input streams from unstable sources without blocking the main processing pipeline is a key requirement for the Data Ingestion Layer.

Algorithmically, the data collection layer implements a producer – consumer pattern, where each source acts as an independent message producer. Data enters the internal event queue:

Algorithm Asynchronous data ingestion:

1. Initialisation of sources (RSS, API, Telegram).

2. Parallel launch of asynchronous collection tasks.
3. Receiving a raw message (raw item).
4. Addition to the processing queue with a timestamp.
5. Monitoring timeouts and request retries.

This approach allows you to isolate failures of individual sources and prevent cascading system failures.

2. Preprocessing Layer

At this level, data is converted to a unified internal format. The main tasks include:

- cleaning text of HTML markup;
- normalising encoding and language;
- extracting embedded media files;
- deduplicating messages.

Deduplication is implemented using a combined approach:

- hashing of normalised text;
- comparison of perceptual image hashes;
- semantic comparison of text embeddings to identify rephrased copies.

3. Modal Analysis Layer

The modal layer consists of three independent subsystems:

- text analysis;
- image analysis;
- video analysis.

Each subsystem functions autonomously and is optimised for its own type of data, allowing the system to be scaled flexibly and individual modules to be replaced without affecting other components.

4. Multimodal Fusion Layer

The fusion layer aggregates the results of different modality analyses. The system supports two strategies:

- early fusion – combining features at the vector representation level;
- late fusion – aggregation of individual probabilities with weighting coefficients.

The final confidence score is calculated as the weighted sum of partial scores:

$$Score = w_t P_{text} + w_i P_{image} + w_v P_{video} \quad (1)$$

5. Decision Layer

The final layer forms:

- an integrated assessment of reliability;
- a list of key factors that influenced the decision;
- explanatory attributes for further audit.

4.3 General architecture of the software system "Vidoglyad"

The developed system is implemented as a desktop application with a graphical interface (WPF, .NET), which provides a modular and extensible architecture. The main functional components of the system are shown in Figure 1 (descriptive):

- media loading and preview module;
- metadata analysis module (EXIF);
- module for searching for similar news in open sources;
- AI-based analytical processing module;
- report generation and results export module.

A key feature of the architecture is the clear separation of responsibilities between modules, which allows individual components to be replaced or supplemented without changing the user interface.

Media content analysis (images and videos).

The system supports common image formats (JPEG, PNG, BMP, GIF) and video formats (MP4, AVI, MKV, etc.). After uploading a file, the user can visually review the material in the program interface.

For images, a secure download mechanism with caching disabled has been implemented to prevent the use of outdated data:

```
var bi = new BitmapImage();
bi.BeginInit();
bi.CacheOption = BitmapCacheOption.OnLoad;
bi.UriSource = new Uri(filePath);
bi.EndInit();
bi.Freeze();
```

This approach ensures that files are displayed correctly and prepared for further analysis.

Obtaining and analysing metadata (EXIF)

One of the key elements of media authenticity verification is metadata analysis. For photos, the system reads EXIF information, including:

- date and time of shooting;
- camera model;
- availability of GPS coordinates;
- comments and service fields.

For video files, where EXIF data is usually absent, the system generates a file information profile based on file system attributes (creation date, size, path).

The metadata obtained is used as additional context for the artificial intelligence analysis module.

Search for similar news in open sources.

A separate system scenario is designed to analyse news materials by URL. In this mode, the system performs an automated search for similar or identical news items in search engines.

For this purpose, a specialised search service is used, which interacts with the API of search platforms (through aggregators, in particular SerpAPI). Example of a search call:

```
var results = await _search.SearchAsync(query, cancellationToken);
```

The result is a list of materials with the following information:

- title;
- source;
- link;
- a short text fragment.

This data allows you to determine the date of the first publication, the geography of the news distribution, and the degree of its replication.

Artificial intelligence analytical module.

The central element of the system is the analytical processing module, implemented through the IAIAnalyzer interface. This allows the use of various AI sources (cloud APIs or local models) without changing the rest of the system.

The analytical query is formed as a structured prompt that clearly defines the role of the model, the list of tasks, and the response format. Separate prompts are used for different scenarios:

- media content analysis (geolocation, visual verification);
- news analysis by URL (source search, fact-checking, manipulation detection).

The analysis is called asynchronously:

```
await _ai.AnalyzeAsync(report, prompt, cancellationToken);
```

The analysis result is stored in a structured report and displayed to the user in text form.

Report generation and export of results. Based on the analysis results, the system generates a summary report containing:

- the results of the AI analysis;
- EXIF data or file information;
- a list of similar news items found.

The results can be exported in HTML format, allowing the reports to be used for further documentation, publication or analytical research.

4.4 Implementation of the Vidoglyad software system in C#/.NET

The system is implemented in C# using standard .NET frameworks. For network calls, a managed HTTP client mechanism is used, which provides control over timeouts and request retries. Data stream processing is implemented based on asynchronous services, which allows for parallel analysis of a large number of sources.

Artificial intelligence modules are integrated in two ways: through external APIs for computationally intensive models and through local execution of models exported to ONNX format. This approach reduces delays and increases system autonomy.

1. Asynchronous processing model

The entire system is built around the async/await asynchronous paradigm. Each logical processing stage is implemented as an independent service that runs in a non-blocking manner in Figure 1.

Figure 1: Example of a logical pipeline

This approach allows thousands of messages to be processed without blocking the UI or server threads.

2. Working with network sources

Effective work with network sources is a critical component of intelligent information flow analysis systems, since it is at this stage that the primary data array is formed for further multimodal processing. In the context of detecting fake news, the system must interact with a large number of heterogeneous sources that differ in access protocols, data formats, response speed, and stability.

To ensure reliable and scalable interaction with such sources, the system uses a managed HTTP client mechanism of the .NET platform. The main principle is to avoid creating disposable clients for each request and to switch to centralised management of the network connection lifecycle.

Timeout control.

Response timeout control is necessary to prevent thread blocking in cases of slow or incorrect operation of external servers. For each source type (RSS, REST API, HTML pages, Telegram gateways), separate timeout values are set, which take into account the characteristic latency of the service and acceptable waiting times.

Within the scope of the programme algorithm, timeout is considered a technical parameter and part of the source access policy. If the permissible feedback time at the data source is systematically exceeded, the system reduces the frequency of requests or temporarily excludes it from the active data collection cycle. This approach allows the pipeline to maintain stable operation even in the presence of unstable or overloaded external resources.

Request retries and recovery policies.

Network failures, temporary server errors, or access speed limitations are common for open information sources. In order to minimise data loss, the system uses a request retry mechanism with a controlled number of attempts and delays between them.

Repeats are implemented taking into account the type of error. For example, in the case of temporary errors (HTTP 5xx), a retry with an exponential delay is allowed, while in the case of client errors (HTTP 4xx), a retry is usually not performed. This approach reduces the load on external services and avoids access blocking due to aggressive client behaviour.

Centralised header configuration scheme.

To increase the transparency of the system, unify requests, facilitate the audit of network interaction, and simplify the process of managing access to resources in the system, a centralised header configuration scheme is implemented, which consists of ensuring the correctness of HTTP header formation (User-Agent, Accept, Authorisation). This scheme enables rapid adaptation to changes in source requirements (for example, when updating access policies or introducing restrictions on the frequency of requests).

Socket exhaustion protection mechanism.

To prevent the problem of socket exhaustion in long-running network applications (which occurs when HTTP connections are not managed correctly), the system uses connection pooling and client lifecycle control [17]. The exhaustion protection mechanism enables: processing thousands of requests over a long period of time without compromising performance; stable execution of continuous monitoring of news feeds and social platforms in Figure 2.

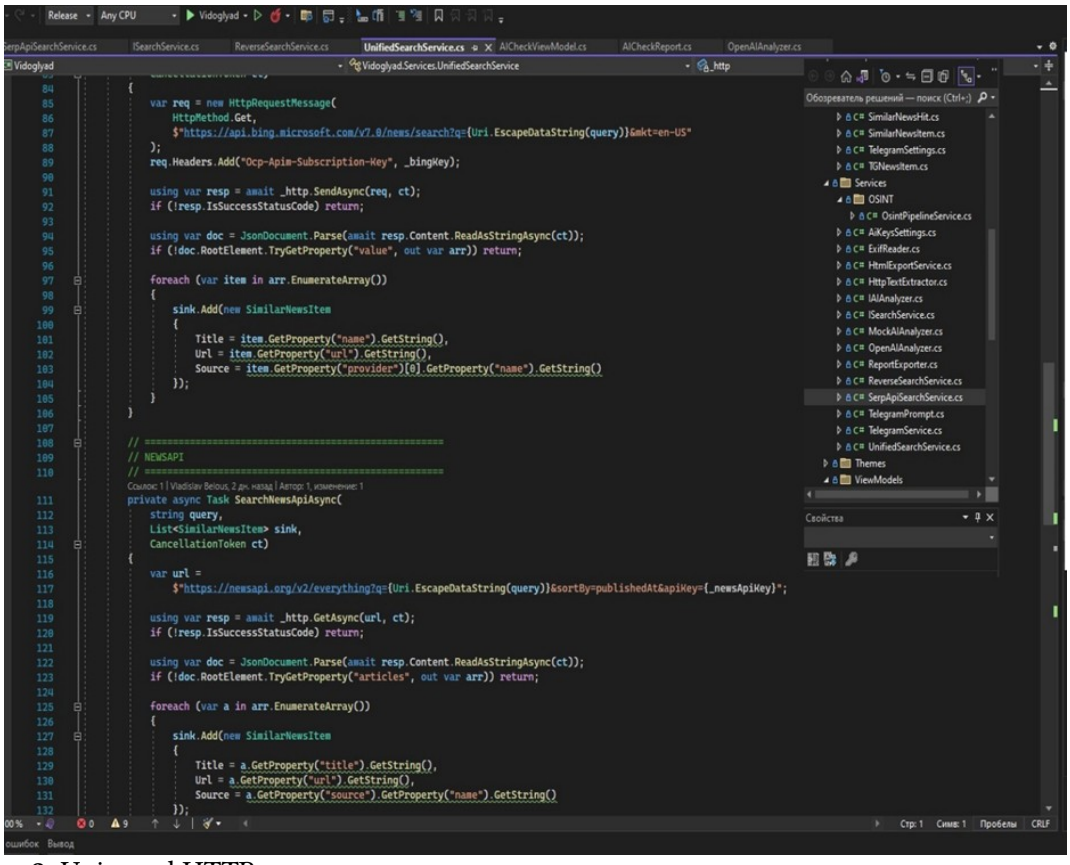


Figure 2: Universal HTTP parser

3. Integration of artificial intelligence modules

In the software product, the integration of artificial intelligence modules is implemented as a separate service layer that provides analysis of text and multimedia content without being tied to the user interface. Interaction with AI is carried out through a unified interface, which allows you to easily change or expand the source of intelligent analysis without modifying the main logic of the application.

The module supports the processing of the following types of input data: URL links; text, graphics, Exchangeable Image File Format (metadata). To analyse multimedia data and reduce the load on the network (ensuring stable operation), graphic images are optimised through compression and scaling. Requests to AI are structured with a clear distinction between permitted and prohibited types of analysis, which minimises the risk of incorrect or speculative conclusions.

To ensure the reliability and security of the system, a continuity function is provided. For example, in the event of external service errors or in the absence of API keys, the system informs the user of the analysis limitations without stopping data processing. The results obtained are processed in a unified manner and can be used both for display in the interface and for generating detailed analytical reports in Figure 3.

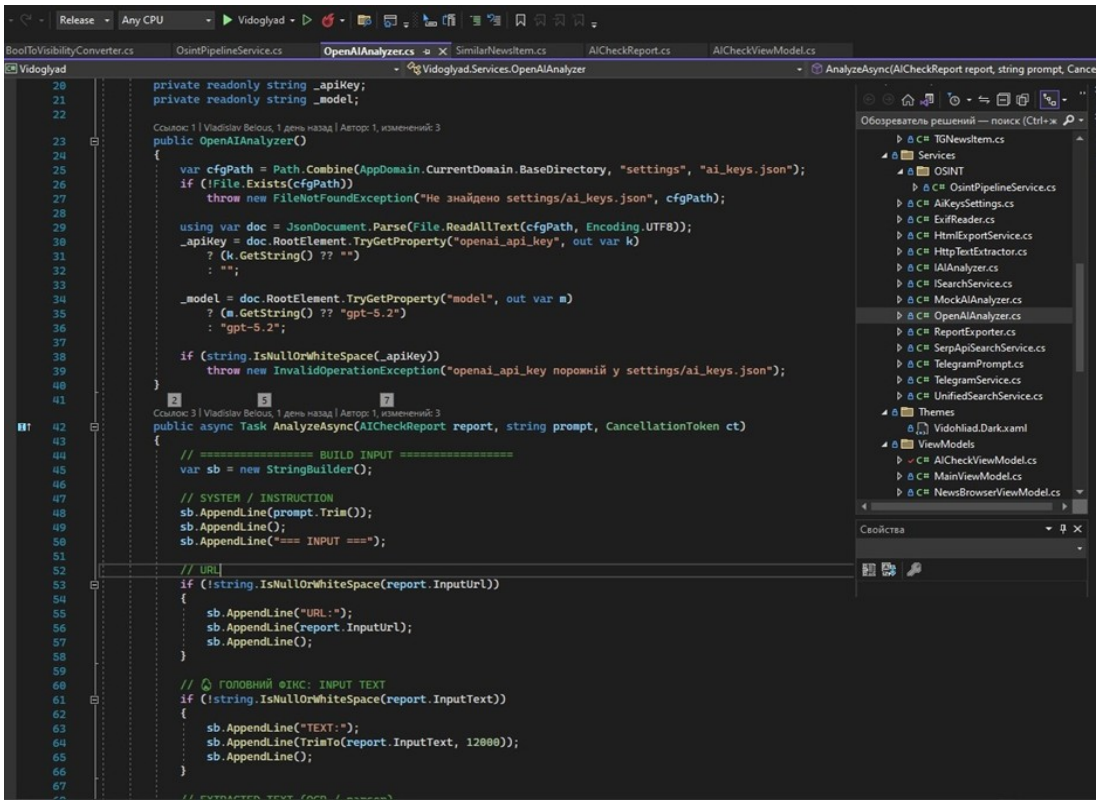


Figure 3: Connector to OpenAI with a private key

The integration of artificial intelligence modules within the system is built with a compromise between the computational complexity of the models, latency requirements, and the need for autonomous operation. To this end, the system supports two complementary inference modes: the use of external APIs and local execution of models in ONNX format.

3.1. Use of external APIs

External APIs are used to access computationally complex models, such as large language models or deep video analysis algorithms. The main advantage of this approach is the ability to scale computations without the need to maintain your own infrastructure.

In practical applications, external APIs are used for tasks that:

- require significant computational resources;
- are not performed in real time;
- allow for some delay in obtaining results.

The disadvantage of this approach is the latency associated with network calls, as well as dependence on the availability of third-party services. Therefore, the system provides mechanisms for caching inference results and reusing already processed data.

3.2. Local inference based on ONNX

For tasks that require low latency or autonomous operation, the system uses local inference of models exported to the ONNX format. This approach allows analysis to be performed without resorting to external services, which is especially important in conditions of limited network access or increased confidentiality requirements.

Local inference is used for:

- text classification;

- detecting image reuse;
- initial assessment of video content.

Running models locally provides control over CPU and GPU usage, allowing the system to be adapted to available hardware resources. In addition, it reduces latency and ensures predictable performance.

Comparative analysis of inference modes

A hybrid integration scheme combines the advantages of both approaches. External APIs provide high accuracy for complex tasks, while local inference ensures a quick system response to new information flows. This approach is the optimal solution for fake news detection systems that need to operate in continuous monitoring mode and ensure a rapid response to the appearance of potentially unreliable content.

4.5 Methods for verifying photo and video content

Verification of photo and video content within the Vidoglyad software, an automated system for detecting fake news, is implemented as a multi-stage process, where each method provides the independent signal of authenticity. This approach reduces the impact of errors in individual algorithms and ensures the system's resilience to complex disinformation scenarios, particularly in cases of deliberate combination of real and synthetic content.

1. Analytical processing of metadata.

At the initial stage of verification, the metadata of media files is analysed. Analytical processing helps to identify basic spatial and temporal inconsistencies. For images and videos, EXIF data is analysed, unless it has been deliberately deleted or altered. The key parameters of the analysis are: geolocation coordinates, file creation timestamps, shooting parameters and device model.

Algorithmically, verification is implemented by comparing metadata with the context of the event in the news. For example, if the text of the message states that the event took place on a specific day or in a specific geographical region, the time and space parameters of the file are compared with these statements. A discrepancy is not automatic proof of a fake, but it does form a negative signal that is taken into account at the multimodal fusion stage.

Particular attention is paid to processing cases of missing or partially damaged metadata. In such situations, the system does not automatically lower the reliability rating, but proceeds to the next levels of analysis, which reduces the number of false positives.

2. Detection of reuse of graphic content.

To detect one of the most common methods of disinformation, the reuse of photos or videos from a different time or in a different context, the system uses the perceptual hashing method (pHash algorithms and its modifications). The difference between perceptual hashing is the stability of the study of minor image changes (compression, scaling, color depth correction). The algorithm forms a compact binary representation of visual content, which is used for comparison with elements of the internal index of previously processed images.

When implementing the Vidoglyad software, this method allows detecting graphic data that previously related to other events but is presented as current. When detecting a similarity that exceeds a specified threshold, the system records the reuse and sends a corresponding signal to the decision-making module.

3. Detection of local image manipulations.

For a more in-depth analysis, methods for detecting local manipulations are used, which are aimed at identifying the editing of individual image fragments. Such manipulations include object insertion, retouching, background replacement, or combining multiple images.

The system uses convolutional neural networks trained on specialised datasets of manipulated images. The main features of the analysis are sharp noise transitions, local compression anomalies, and unnatural object boundaries. The algorithm sequentially reviews local areas of the image using the "Sliding Window" procedure and builds a map of the probability of manipulation. The obtained

indicators are not used separately, but are reduced to an integral index that takes into account both the area of potentially altered zones and the degree of their abnormality.

Within the framework of visual content analysis, special attention is paid to checking images for hidden interventions and distortions. In the context of information warfare, even minor local edits can significantly change the perception of an event, so automated detection of manipulations is a critically important element of OSINT analysis. For this purpose, the system implements a module for in-depth analysis of local image changes, which complements classic visual and metadata verification in Figure 4.

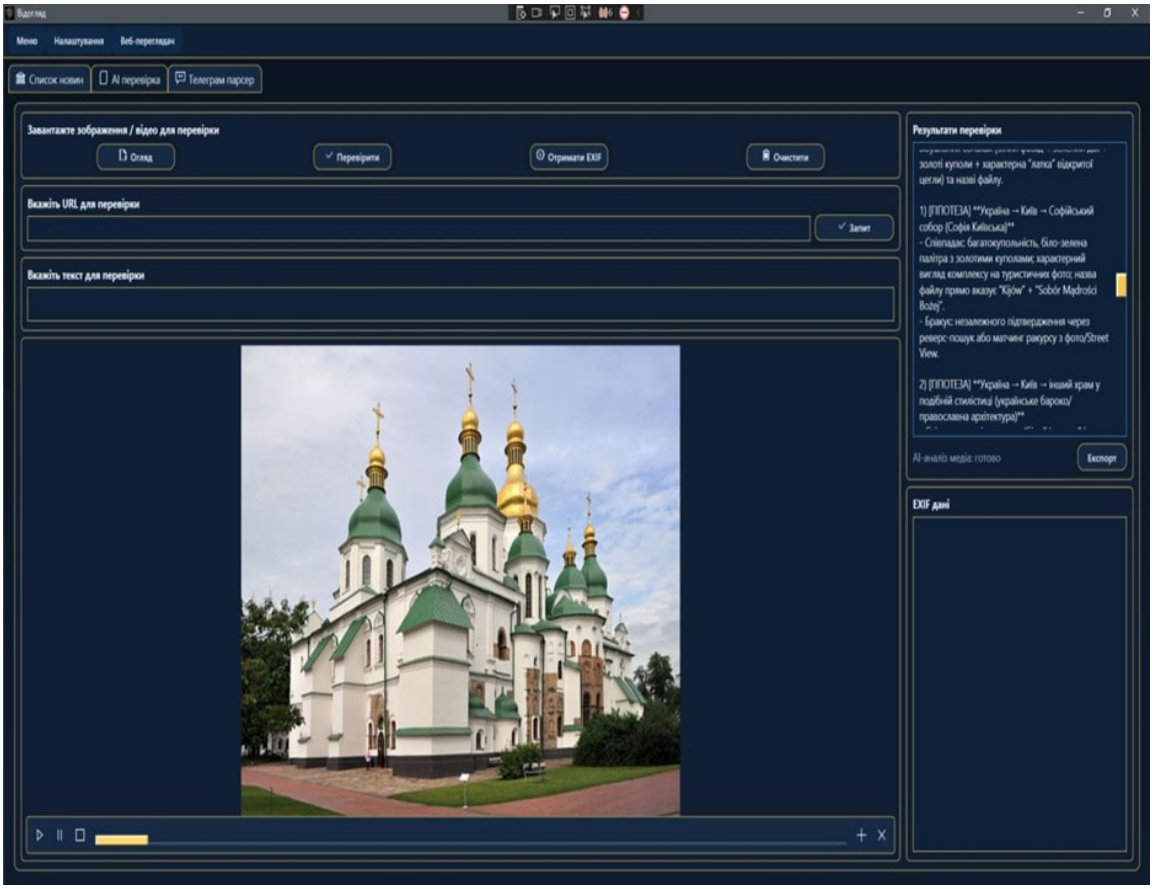


Figure 4: Image search query and analysis results

4. Video content analysis and deepfakes.

Video verification is the most difficult and resource-intensive stage due to the need to simultaneously analyse the spatial and temporal characteristics of the source data. First, key frames are extracted, which reduces the computational load without significantly losing the database under investigation.

To detect synthetic videos, frame time sequences are analysed, for example, by performing lip-syncing (aligning lip movements and facial micro-expressions with audio). Even with high-quality synthesis, these features are sensitive to artefacts of generative models. The process of combining several independent indicators plays a significant role, allowing deepfake content to be detected in the absence of obvious visual defects.

4.6 Monitoring Telegram and news feeds

Due to the lack of centralised editorial moderation, high publication speed, and the possibility of anonymous channel administration, Telegram is one of the key channels for spreading disinformation in the modern information environment. This necessitates continuous automated monitoring of both text and multimodal content.

1. Monitoring architecture.

Telegram monitoring is implemented as a separate sub-system integrated into the overall data processing pipeline. Its architecture is based on asynchronous collection of messages from a specified list of public channels and groups. Each message is stored along with timestamps, source ID, embedded media content, and metadata. This approach creates an archive of messages that is used for further analysis and search for recurring patterns. In the Vydoglyad software architecture, the monitoring module is separated from the analysis modules, which allows data collection to be scaled independently of the computational costs of inference in Figure 5.

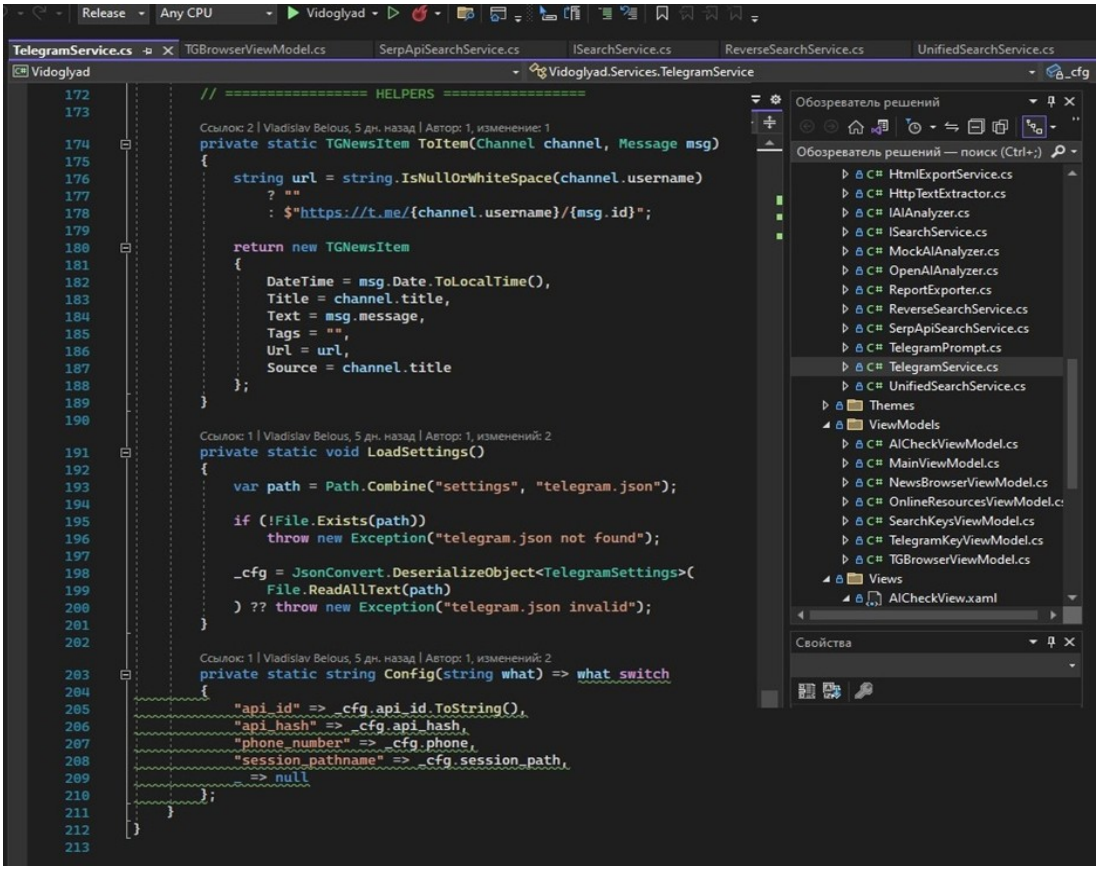


Figure 5: TelegramService code for data retrieval

2. Researching the dynamics of narratives.

The collected data is used to study the temporal dynamics of information narratives. To facilitate the study, a diffusion graph is formed, where the vertices are messages or channels, and the edges are temporal and semantic connections between them. The semantic similarity of texts is assessed using vector representations, which allow identifying paraphrased versions of the same message.

Analysis of the graph makes it possible to identify the primary sources of fake messages and track their further dissemination through the network of channels. Particular attention is paid to identifying coordinated information campaigns, which are characterised by the synchronous publication of similar content on several channels with a minimal time lag.

3. Example of a usage scenario.

In a typical scenario, the same text or image appears on several Telegram channels with a time lag of several minutes or hours. The system detects a high level of semantic and visual similarity, builds a distribution chain, and increases the integral risk indicator of unreliability. If, at the same time, the reuse of visual content or inconsistency with the event context is detected, the decision about potential fakeness is reinforced in Figure 6.

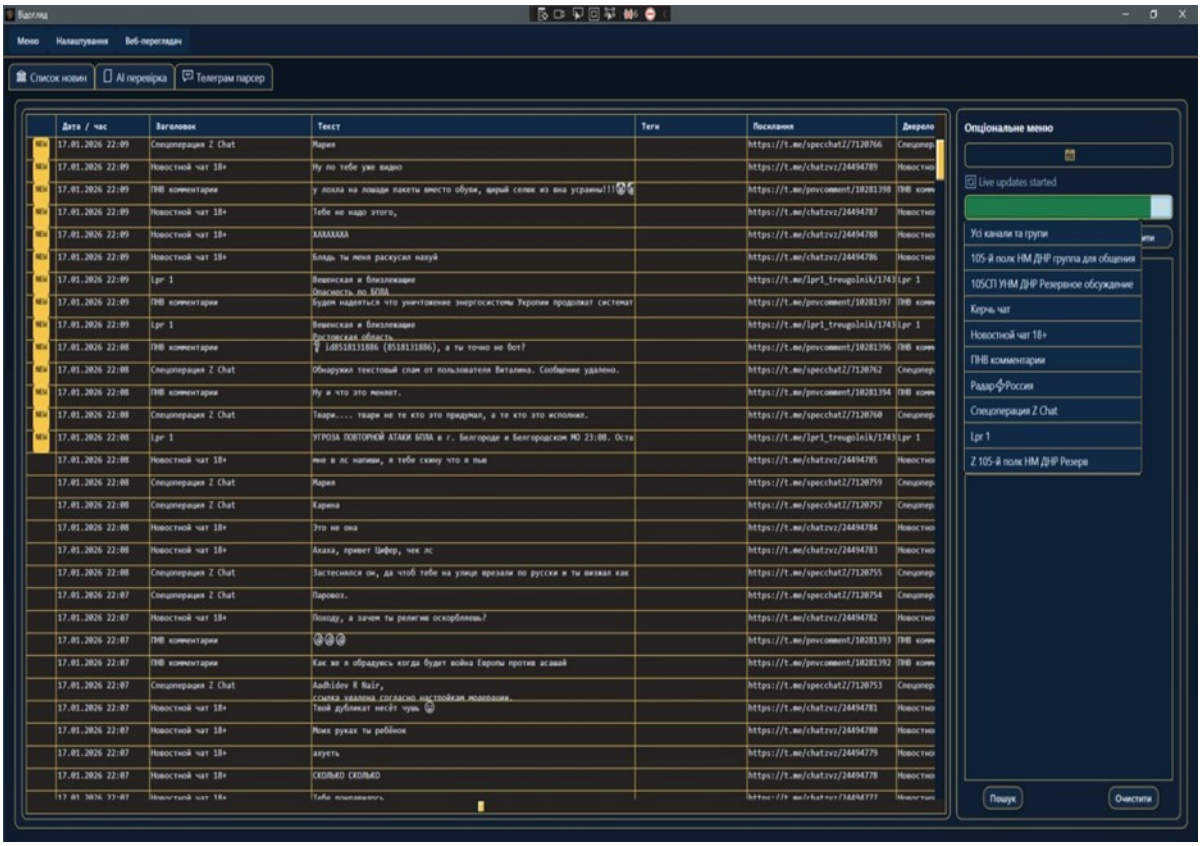


Figure 6: Parsing channel results in Live mode

Sub-item 4 News feeds based on RSS feeds (screenshot + explanation).

In the programme, the RSS module is implemented as a separate news gathering service that periodically accesses specified RSS sources and forms a unified stream of news items (NewsItem). Each item includes basic fields: title, short description, publication date, source, and link to the full article.

This approach allows you to:

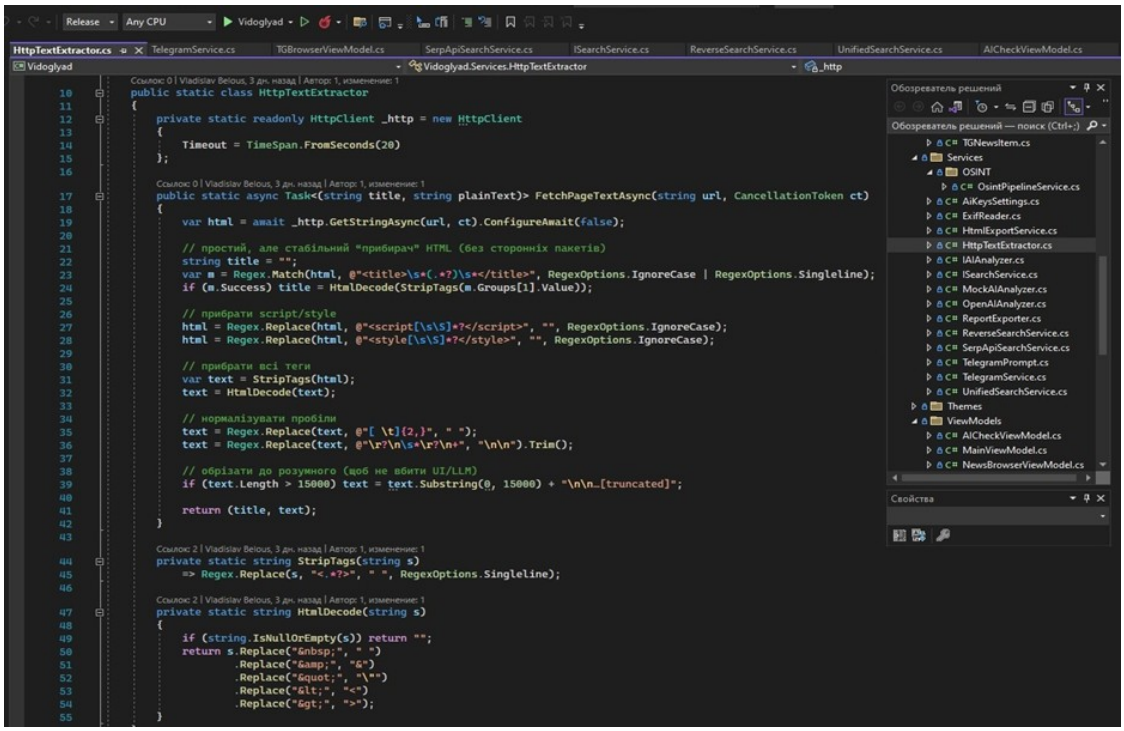
- aggregate news from different media into a single data model;
- perform further filtering by time, source or topic;
- store the original links for verification of the primary sources.

After collecting RSS data, the system applies filtering and sorting logic at the ViewModel level. This allows the user to:

- view news chronologically;
- separate materials from specific sources;
- respond quickly to new publications.

The HttpTextExtractor module is designed to automatically extract text content from web pages for further analysis. It downloads the HTML code from the specified URL, cleans it of scripts, styles and markup, and converts the page into readable plain text. The page title (<title>) is extracted separately, allowing you to quickly identify the source of the material.

The resulting text is normalised and limited in size, which ensures stable operation of the interface and correct data transfer in the artificial intelligence module. Thus, the component acts as an intermediate link between external web resources and the analytical processing system in Figure 7.



```
public static class HttpTextExtractor
{
    private static readonly HttpClient _http = new HttpClient
    {
        Timeout = TimeSpan.FromSeconds(20)
    };

    public static async Task<(string title, string plainText)> FetchPageTextAsync(string url, CancellationToken ct)
    {
        var html = await _http.GetStringAsync(url, ct).ConfigureAwait(false);

        // простий, але стабільний "прибирач" HTML (без сторонніх пакетів)
        string title = "";
        var m = Regex.Match(html, @"<title>\s*(.*?)\s*</title>", RegexOptions.IgnoreCase | RegexOptions.Singleline);
        if (m.Success) title = HtmlDecode(StripTags(m.Groups[1].Value));

        // прибрати script/style
        html = Regex.Replace(html, @"<script[\s\S]*?</script>", "", RegexOptions.IgnoreCase);
        html = Regex.Replace(html, @"<style[\s\S]*?</style>", "", RegexOptions.IgnoreCase);

        // прибрати всі теги
        var text = StripTags(html);
        text = HtmlDecode(text);

        // нормалізувати пробіли
        text = Regex.Replace(text, @"[ \t]{2,}", " ");
        text = Regex.Replace(text, @"\r?\n\s*\r?\n+", "\n\n").Trim();

        // обрізати до розумного (щоб не вбити UI/LLM)
        if (text.Length > 15000) text = text.Substring(0, 15000) + "\n\n…[truncated]";

        return (title, text);
    }

    private static string StripTags(string s)
        => Regex.Replace(s, "<.*?>", " ", RegexOptions.Singleline);

    private static string HtmlDecode(string s)
    {
        if (string.IsNullOrEmpty(s)) return "";
        return s.Replace(" ", " ")
                .Replace("&amp;", "&")
                .Replace("&quot;", "\"")
                .Replace("&lt;", "<")
                .Replace("&gt;", ">");
    }
```

Figure 7: Code for obtaining the text content of RSS feeds

The code implementation is based on the use of ObservableCollection collections, which ensures instant interface updates without reloading the entire feed.

A key feature of the approach is the integration of RSS news with subsequent analysis modules. The selected news item can be:

- transferred to the AI verification module for authenticity assessment;
- used as input data to search for similar publications on the network;
- involved in OSINT analysis of information campaigns.

Thus, the RSS feed is not the end goal, but the entry point of the analytical pipeline in Figure 8.

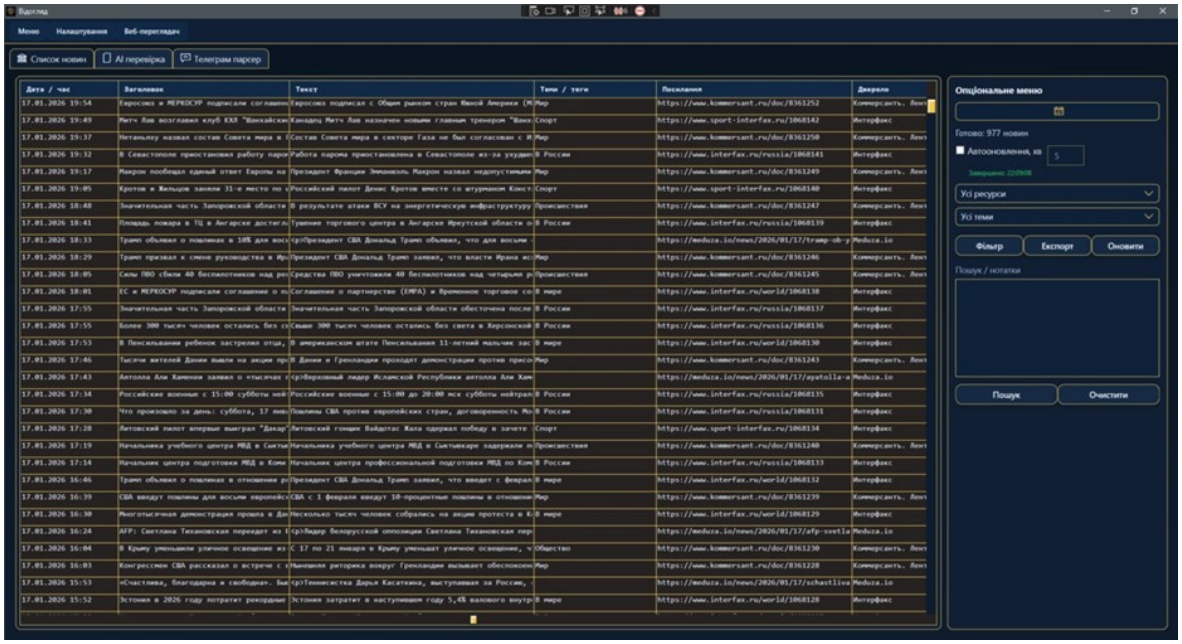


Figure 8: GUI result of obtaining RSS data

The Vidoglyad software product was tested to verify the performance, stability, and analytical efficiency of the implemented modules in real-world conditions of processing news, text, and multimedia data. The main focus was on the modules for news gathering (RSS), AI content verification, EXIF metadata analysis, information matching, and Telegram parsing.

6.1. Testing the news feed module (RSS)

During testing of the news gathering module, the operation of several RSS feed sources (including news agencies and news portals) was verified. The programme correctly:

- downloaded news in the background;
- structures data by date, time, headline, text, source, and link;
- provides filtering by topic and source;
- supports exporting results in tabular form.

The interface screenshots show stable display of over 900 news items without loss of performance, which indicates effective work with data collections and asynchronous update mechanisms.

6.2. Testing the AI module for checking news and texts

The AI module was tested on news reports from various sources, including those with potentially manipulative or propagandistic content. The testing showed that the system:

- performs semantic analysis of text;
- generates a structured OSINT report with information classification;
- detects logical contradictions, lack of confirmation or incomplete data;
- correctly distinguishes between confirmed, partially confirmed and unconfirmed statements.

In practical examples (screenshots with analysis results), the AI module correctly identified information based on only one source and indicated the need for additional verification, which is in line with the principles of the classic OSINT approach.

6.3. Testing the analysis of multimedia content (images and EXIF)

A separate stage of testing was devoted to image analysis. The programme successfully:

- downloaded photos and videos from a local storage device;
- determined the type of media;
- read EXIF metadata (device model, shooting time, file creation time);
- detect discrepancies between the shooting date and the file creation date.

The test scenarios recorded a typical situation of copying or resaving a file, which was correctly interpreted by the system without false claims of falsification. At the same time, the absence of GPS coordinates was clearly indicated in the final conclusion, confirming the correct logic of the analysis restrictions.

6.4. Testing the search for information matches

The search module was tested based on URL links and text queries. The programme:

- generates queries to search engines;
- aggregates results;
- uses AI to interpret the context found.

In cases where there were no supporting sources, the system did not make unfounded conclusions, but clearly indicated the limitations of the available data.

6.5. Overall assessment of results

The test results confirmed that the Vidoglyad software product:

- works stably with large amounts of information;
- correctly implements the principles of OSINT analysis;
- combines automated data collection with analytical interpretation;
- does not go beyond the scope of the data provided, which is critical for the reliability of conclusions.

The results obtained indicate that the programme is ready for practical use in the analysis of news flows, information security and media verification in the context of information warfare.

5 Conclusions

The study thoroughly examined multimodal artificial intelligence approaches for the automated detection of fake news and verification of media content in the modern digital information environment, which is characterised by the high speed of information dissemination, fragmentation of sources, and active use of manipulative technologies. It has been established that the use of isolated analysis methods (only textual or only visual) is insufficient for reliable assessment of information in the context of hybrid information threats.

The results of the study give reason to believe that in order to detect fake news (semantic distortions of text) detecting hidden manipulations (contextual inconsistencies between different modalities, local changes in graphics), it is advisable to use comprehensive processing of text, graphic and video materials within an intelligent system by combining methods of natural language processing, computer vision and video data analysis.

Architectural solutions for collecting, pre-processing, and analysing data from RSS news aggregators and Telegram channels, which form a significant part of the operational information flow, have been analysed. A multi-level architecture of the Vidoglyad intelligent software system is proposed, consisting of five logically separate levels: Data Ingestion Layer, Preprocessing Layer, Modal Analysis Layer, Multimodal Fusion Layer, and Decision Layer. This architecture ensures scalability, modularity, and the possibility of independent development of individual system components.

The feasibility of using multimodal algorithms to ensure comprehensive media verification by checking the authenticity of text, photo, and video materials, including methods for detecting manipulation, retouching, editing, and deepfake content, is revealed. An analytical approach to evaluating the effectiveness of multimodal systems using classic machine learning metrics (precision, recall, F1-score) is justified to perform an objective assessment of the accuracy, completeness, and reliability of automated detection of fake news.

The author's software "Vidoglyad" is proposed, which is a full-fledged OSINT tool for automated detection of fake news and verification of media content based on artificial intelligence methods. The functionality of Vidoglyad has been demonstrated, which provides a comprehensive approach to the analysis of information materials, in particular: multidimensional fact-checking and detection of signs of manipulative influence; data collection from various open sources; automatic verification of content authenticity; assessment of the likelihood of manipulation; presentation of results in a user-friendly format. The Vidoglyad functionality is based on: the use of modules for working with text, graphic and video materials; integration with third-party service APIs and a flexible artificial intelligence service layer (the system has the ability to expand data sources and increase the depth of analysis).

The Vidoglyad software is implemented in real time and demonstrates the effectiveness of the proposed algorithmic solutions and confirms their suitability for practical application. It has been proven that the use of artificial intelligence tools in combination with OSINT approaches allows: reducing the time required to analyse media content; increasing the accuracy of detecting fake news; improving the quality of the information verification process. The proprietary software "Vidoglyad" can be used as a practical tool for: improving information hygiene; countering information threats; supporting informed decision-making in the field of media analysis and cybersecurity.

Prospects for further research lie in expanding the functionality of the Vidoglyad software by training adaptive multimodal models capable of self-learning based on streaming data from various OSINT sources and its scientific and methodological interpretation in the context of information conflicts and cyber threats.

Declaration on Generative AI

While preparing this work, the authors used the AI programs Grammarly Pro to correct text grammar and Identific to search for possible plagiarism. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the publication's content.

References

1. N. Capuano et al., Content-Based Fake News Detect. with Machine and Deep Learning: a Syste-matic Review. Neurocomputing, vol. 530 (2023) 91–103. doi: 10.1016/j.neucom.2023.02.005
2. E. Saquete, et al., Fighting Post-truth Using Natural Language Process.: a Review and Open Challenges, Expert Systems with Applications, 141 (2020) 112943. doi: 10.1016/j.eswa.2019.112943
3. F. Sadeghi, A. Bidgoly, H. Amirkhani, Fake News Detect. on Soc. Media using a Natural Language Inference Approach, Multimed. Tools and Applications, 81 (2020) 33801–33821, doi: 10.1007/s11042-022-12428-8
4. E. Aïmeur, S. Amri, G. Brassard, Fake News, Disinformation and Misinformation in Soc. Media: a Review, Soc. Network Analysis and Mining, 13 (2023) 30. doi: 10.1007/s13278-023-01028-5
5. B. Palani, S. Elango, V. Vignesh, CB-Fake: a Multimodal Deep Learning Framework for Automatic Fake News Detect. using Capsule Neural Network and BERT, Multimedia Tools and Applications, 81 (2021) 5587–5620. doi: 10.1007/s11042-021-11782-3
6. E. Hashmi, et al., Advancing Fake News Detect.: Hybrid Deep Learning with FastText and Explainable AI, IEEE Access 12 (2024) 44462–44480. doi: 10.1109/access.2024.3381038
7. S. Khan, et al., Online Multimedia Verification with Computational Tools and OSINT: Russia-Ukraine Conflict Case Studies, ArXiv, (2023). doi: 10.48550/arxiv.2310.01978
8. V. Bilous, et al., Open Source Intelligence for War Crime Documentation, Workshop Cybersecur. Providing in Inf. and Telecommun. Systems, 3654 (2024) 368–375
9. V. Bilous, et al., Cyber Evidence Software as the Digital Forensics Tools in the Investigation of Cybercrime, Workshop Cybersecurity Providing in Inf. and Telecommun. Systems, 3991 (2025) 26–37
10. J. Evangelista et al., Systematic Literature Review to Investigate the Application of Open Source Intelligence (OSINT) with Artificial Intelligence, Journal of Applied Secur. Research, 16 (2020), 345–369. doi: 10.1080/19361610.2020.1761737
11. M. Al-Alshaqi, D. Rawat, C. Liu, Ensemble Techniques for Robust Fake News Detect. in: Integrating Transformers, Natural Language Processing, and Machine Learning. Sensors 24 (2024) 6962. doi:10.3390/s24186062
12. S. Mahadevkar et al., A Review on Machine Le12arning Styles in Computer Vision – Techniques and Future Directions, IEEE Access, 10 (2022) 107293–107329. doi: 10.1109/ACCESS.2022.3209825
13. M. Astafieva, et al., Formation of High School Students' Resistance to Destructive Inf. Influences, in: Cybersecur. Providing in Inf. and Telecommun. Systems(CPITS), vol. 3421 (2023) 87-96.
14. M. Astafieva, et al., Organization of Independent Work of Students in LMS Moodle using a Metacognitive Approach (on the Ex. of Physical and Mathematical Disciplines), in: International Conference on Inf. and Commun. Technologies in Education, Research, and Industrial Applications (ICTERI), vol 1980 2023, 303-312. doi: 10.1007/978-3-031-48325-7_23
15. X. Zhao, et al., A Review of Convolutional Neural Networks in Computer Vision. Artificial Intelligence Review 57 (2024) 99. doi: 10.1007/s10462-024-10721-6
16. J. Dong, et al., Dual Encoding for Video Retrieval by Text. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44 (2021) 4065–4080. doi: 10.1109/tpami.2021.3059295
17. L. Ju, et al., Using Asynchronous Frameworks and Database Connect. Pools to Enhance Web Application Performance in High-Concurrency Environments. 2024 in 8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC) (2024) 742–747. doi: 10.1109/i-smac61858.2024.10714639