DOI 10.31891/2307-5732-2024-343-6-6
УДК 004.056:004.89:621.398:658.239

БУРОВ ЄВГЕН
Національний університет «Львівська політехніка»
https://orcid.org/0000-0001-8653-1520
e-mail: yevhen.v.burov@lpnu.ua
ЖОВНІР ЮРІЙ
Національний університет «Львівська політехніка»
https://orcid.org/0009-0006-6186-2861
e-mail: zhovnir@astra.in.ua
ЗАХАРІЯ ОЛЕГ
Національний університет «Львівська політехніка»
https://orcid.org/0009-0008-6979-129X
e-mail: ozakhar@gmail.com

БАЧЕННЯ ТА ВПРОВАДЖЕННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ БЕЗПЕКИ

Впровадження штучного інтелекту в усі сфери технологій обіцяє підвищення продуктивності. Системи безпеки з високим рівнем непередбачуваності, ризиків, вимогою діяти на випередження та передбачати розвиток інцидентів належать до сфери, де використання штучного інтелекту може принести значну віддачу від інвестицій. У цій статті описано бачення впровадження інтелектуальної системи безпеки. Така система розглядається як ситуаційно-орієнтована система, яка виявляє ситуації та обробляє їх розвиток у реальному часі. Система використовує онтологію, отриману від GFO, для створення загальної специфікації понять і об'єктів. Початковий варіант системи безпеки містить фронт-енд і серверну частини, що реалізують управління контролем доступу і спостереження. База знань використовує сценарії типових інцидентів і протоколи їх обробки для визначення роботи системи.

Ключові слова: штучний інтелект, інтелектуальна система безпеки, ситуаційно-орієнтовані системи, управління контролем доступу, відеоспостереження, база знань, протоколи обробки інцидентів, онтологія GFO, автоматизація процесів безпеки.

BUROV YEVHEN, ZHOVNIR YURIY, ZAKHARYA OLEH
Lviv Polytechnic National University

THE VISION AND IMPLEMENTATION OF INTELLIGENT SECURITY SYSTEM

The implementation of artificial intelligence in all areas of information technology promises significant productivity improvements, particularly in security systems. In the context of modern threats and a high level of unpredictability, security systems are an environment where the use of intelligent technologies can provide maximum efficiency. Intelligent security systems not only respond to incidents but also predict their development, enabling proactive actions and risk minimization. Such systems represent a new approach to security, focusing on situational awareness and real-time data processing.

This article presents approaches to the implementation of an intelligent security system capable of detecting and analyzing situations, using the collected information to make operational decisions. The core of the system is based on the General Formal Ontology, which allows for the creation of a unified model of concepts and objects, ensuring data consistency and interpretability. This approach simplifies the integration of the system with other components and expands its application capabilities.

The initial version of the intelligent security system includes two key modules: a front-end for user interaction and a back-end that handles data processing and system management. These components perform essential functions such as access control and surveillance, allowing for effective management of security processes in real-time. A vital part of the system is the knowledge base, which contains scenarios of typical incidents and algorithms for handling them, enhancing the responsiveness and reducing the number of false alerts.

The system provides not only monitoring but also analytics, optimizing risk management processes. The implementation of such solutions opens new perspectives for the integration of AI into security systems, enhancing their efficiency. The proposed architecture allows for the development and scaling of the system in accordance with new requirements and challenges faced by modern infrastructures. This research represents a significant step towards the creation of next-generation security solutions, focused on high performance, reliability, and situational awareness.

Keywords: artificial Intelligence, intelligent security system, situation-oriented systems, access control management, video surveillance, knowledge base, incident handling protocols, General Formal Ontology, security process automation

Statement of the problem in a general form and its connection with important scientific or practical tasks

The core trend in the development of information systems today is it introduction of artificial intelligence (AI) in all areas of our life. The success of generative AI, using large language models provides the substantial increase in human productivity providing simplified access to information and knowledge, creating content, learning and reasoning.

Security systems are the important field for the application of AI, because of the need to quickly assess situations and make real-time decisions in a variety of contexts, sometimes quite unpredictable. In security it is hard to describe beforehand all possible usage scenarios. Therefore, it makes sense to consider the intelligent security system as situation- aware system, able to monitor the environment, anticipate the changes in it, analyze and reason about them. The implementation of intelligent security system requires both reactive and proactive behavior, which adds to the challenges of creating situation-aware system, such as using experiential, contextual knowledge, acting in real -time, assessing the results and updating the knowledge base.

Taking in consideration the inherent complexity of creating situation-aware systems and following the idea of gradual approach to product development, we envisage the creation of intelligent security system as an

evolutionary, step-by-step process, starting with minimal viable product, obtaining feedback from its usage and adding new features, based on it. However, while moving through iterations in the development, it is useful to have a global, strategic vision of intelligent security system in order to cut the technical debt and understand the direction for product development.

This article aims to present both the vision and the first iteration of security system implementation for large residential community, describing the relationships between them. The article is structured as follows. In the Background section we analyze the status in the introduction of AI into security systems, focusing on access control and surveillance functions. In the Methods and materials part, we present our vision of intelligent security system and in the Results part we describe the first iteration of security system. The article is concluded with Discussions and conclusions section in which we discuss the next steps for development, the introduction of the new intelligent features.

Analysis of recent research and publications

In the recent years the interest into the introduction of intelligent features in security systems is growing [1]. The trends in intelligent security systems focus on leveraging AI techniques such as machine learning, deep learning, and natural language processing to enhance security. These methods facilitate automated and intelligent security processes, improving the detection and response to security threats. Additionally, knowledge representation and reasoning, along with rule-based expert systems, are being integrated to create more adaptive and effective security solutions.

The work [2] presents a survey about the impact of artificial intelligence on data system security. The authors note that, on the one hand, companies are experiencing difficulty in dealing with security challenges with regard to a variety of issues ranging from system openness, decision making, quality control, and web domain. On the other hand, in the last decade, research has focused on security capabilities based on tools such as platform complacency, intelligent trees, modeling methods, and outage management systems to understand the interplay between AI and those issues.

One of the promising areas of application for intelligent security solutions are intelligent residences. In the article [3] is stated that traditional security solutions cannot be applied to smart cities, because of the heterogeneity, scalability, and dynamic characteristics of such cities. [Cui] survey the current stage of smart cities development with respect to security and privacy to provide an overview of both the academic and industrial fields and to pave the way for further exploration.

The intelligent security system gets its data from the variety of sources, such as sensors, surveillance cameras etc. The intelligent data collection and pre-processing start getting more attention in the research community.

Recently, the developments in smart sensors for intelligent environments have been marked by significant advancements in sensor technology, data processing, and integration with artificial intelligence. These innovations are transforming how environments are monitored and managed, offering enhanced capabilities for a variety of applications. Smart sensors have evolved to include integrated intelligent abilities, such as fiber-optic structures and chemical sensing materials, which enhance their performance and functionality. These sensors are organized with distributed sensing nodes, allowing for comprehensive data collection and analysis at the system level, which is crucial for complex environments [4]

Intelligent sensing, which combines AI with smart sensors, can autonomously solve complex problems. AI-based algorithms enhance the capabilities of smart sensors, enabling real-time monitoring and decision-making in intelligent security systems. This integration provides promising solutions for a wide range of applications, from natural language processing to computer vision [5]. Overall, the integration of smart sensors with AI and IoT technologies is paving the way for more efficient and sustainable intelligent systems, addressing challenges in data acquisition and processing while offering new opportunities for innovation and application [6].

Intelligent sensing is performed by a network of interacting intelligent agents. An intelligent agent is a computational system or entity that possesses the capability to perceive its environment, process information, make decisions, and take actions to achieve specific goals or objectives [7]. Intelligent agents are capable of autonomy, obtaining the information from the world using sensors. They can make decisions, act proactively and learn.

In the article [8] authors state that intelligent agents can be deployed in scenarios like unmanned automated systems, electrical distribution grids, communication networks in space, and large-scale computational arrays. Their ability to collaborate, share information, and adapt to changing environments makes them valuable assets in enhancing the security and surveillance capabilities of systems in diverse settings. The authors analyze several scenarios to consider the types of threats intelligent agents might be explaining the importance of this field and the motivation for its emergence, where a software agent resides on a system, and is responsible for defending the system from cyber compromises and enabling the response and recovery of the system, usually autonomously expected to encounter and what actions would potentially be beneficial for them to take in response.

In [9] the concept of autonomous intelligent cyber-security agent is introduced. The importance of this field and the motivation for its emergence are explained. The software agent resides on a system and is responsible for defending the system from cyber compromises and enabling the response and recovery of the system, usually autonomously.

Machine learning is a promising part of intelligent security systems, providing the ability to learn the new and update existing threat patterns from experience. An article [10] presents an example of intelligent security system based on deep learning. An intelligent security system based on two methods for identifying persons, gait recognition

and face recognition, was described. The two methods mentioned above can be used to give persons access to different buildings, facilities, institutions, etc. or to different indoor or outdoor areas. The system uses face and gait recognition to identify people who should be granted access to a facility or a specific area within the facility. Two deep learning models for gait and face recognition have been developed and described.

The work [11] explores the application of deep learning models for anomaly detection and face recognition in IoT devices within the context of smart homes. It considers the six models, namely, LR-XGB-CNN, LR-GBC-CNN, LR-CBC-CNN, LR-HGBC-CNN, LR-ABC-CNN, and LR-LGBM-CNN, and evaluate their performance. The models were trained and tested on labeled datasets of sensor readings and face images, using a range of performance metrics to assess their effectiveness. Performance evaluations were conducted for each of the proposed models, revealing their strengths and areas for improvement.

Another work dedicated to anomaly detection in surveillance system is [12]. The authors stress the importance of understanding the monitored environment. This includes defining and analyzing normality concepts, such as normal paths and behaviors, to detect anomalies effectively. The proposed framework allows for the treatment of uncertainty and can be generalized across various surveillance domains. By instantiating these normality concepts in real environments, the system can provide high-level information about object behaviors, facilitating informed decision-making in surveillance operations.

Intelligent access control and management systems are evolving with the integration of advanced technologies such as artificial intelligence (AI), Internet of Things (IoT), and blockchain. These systems aim to enhance security, efficiency, and user convenience by leveraging these technologies in innovative ways.

The work [13] uses emotion recognition to assess the visitor intentions. This approach allows the system to identify potentially aggressive visitors and alert security personnel accordingly. Additionally, the AI can flag individuals stored in special databases, ensuring that unwanted visitors receive increased scrutiny. The paper also discusses the training algorithm for a neural network, which is essential for improving the accuracy and effectiveness of these intelligent access control systems.

In the work [14] an IoT-based access management system is presented. It integrates Artificial Intelligence (AI) and Blockchains (BC) for intelligent access control and utilizes facial detection and recognition through a Multi-Task Cascaded Convolutional Network (MTCNN) and a Convolutional Neural Network (FRMN) for real-time identification. Access rights are determined automatically without human intervention, with decisions and images stored using HyperLedger Fabric (HLF). This approach enhances security, transparency, and efficiency in access management by leveraging the strengths of AI and BC technologies.

The article [15] proposes a risk-based access control model for IoT technology that considers real-time data information request for IoT devices and gives dynamic feedback. The model uses IoT environment features to estimate the security risk associated with each access request using user context, resource sensitivity, action severity and risk history as inputs for security risk estimation algorithm that is responsible for access decision. After this, the model uses smart contracts to provide adaptive features in which the user behaviour is monitored to detect any abnormal actions from authorized users.

The work [16] argues that traditional access control mechanisms are not expressive enough to handle complex access control needs. The authors propose the creation of the system, that builds upon existing work in attribute based access control model, captures physical context collected from sensed data (attributes), and performs dynamic reasoning over these attributes and context driven policies using Semantic Web technologies to execute access control decisions. Reasoning over user context, details of information collected by cloud service provider and device type the proposed method generates access control decisions. Developed access control system is supplemented by another sub-system that detects intrusions into smart home based on both network and behavioral data. The combined approach serves to determine indicators that a smart home system is under attack, as well as limit what data breach such attacks can achieve.

Intelligent surveillance systems are evolving rapidly, integrating advanced technologies to enhance security and monitoring capabilities. These systems leverage a combination of thermal imaging, high-resolution video, edge computing, multi-sensor integration, and dense sensor networks to provide comprehensive surveillance solutions.

Smart surveillance systems utilize thermal imaging to detect the presence and attributes of objects in a scene. This allows for intelligent control of illumination, activating light sources only when necessary, and optimizing the type and angle of light used [17]. Such systems can selectively operate visible/NIR light cameras to capture images of interest, enhancing the efficiency and effectiveness of surveillance operations [17].

High-resolution video systems employ mega-pixel cameras to capture detailed images across multiple locations simultaneously. These systems provide continuous 360-degree viewing and digital zoom capabilities, ensuring important details are scrutinized while less critical areas are imaged at lower resolutions [18].The feedback control subsystem dynamically allocates resources, optimizing the surveillance process by focusing on areas of interest [18].

Edge computing extends surveillance capabilities by performing computations near the data source, reducing latency in real-time applications. AI and machine learning algorithms, such as Harr-Cascade and lightweight CNNs, are used for efficient object detection and tracking on edge devices [19]. These systems are validated using real-world data, demonstrating their ability to track humans accurately with minimal resource consumption [19].

Some intelligent surveillance systems provide multi-sensor Integration of heterogeneous information from video, audio, and other sensors to detect intrusions. A rule-based model processes this data, generating alarms and

notifications in real-time is presented in [20]. The system's ontology allows for customizable intrusion definitions, adapting to various scenarios and enhancing security measures [20].

Another solution, contributing to the timely detection of threats and increased coverage is using dense sensor networks [21]. Surveillance platforms deploy large numbers of inexpensive sensors to maximize coverage while minimizing costs. These sensors use motion detection and tracking algorithms to convert video signals into motion parameters. The platform's control center employs AI strategies for alarm detection, suitable for various domains such as traffic surveillance and perimeter security.

Highlighting previously unresolved parts of the general problem, to which the article is devoted

Software development process for any complex product should be guided by long-term vision, specifying the features and the strategy for product development. This vision serves as a reference point allowing to compare the current state of the product with ideal, vision state, finding the gaps and planning of closing them in the future stages of development.

This idea is supported by numerous research articles and practical experience in the domain of software development. For example, article [22] says that a well-defined long-term vision is crucial for sustaining software quality and adaptability over time, suggesting that strategic planning can mitigate risks associated with software obsolescence and technical debt.

The article [23] explores the dichotomy between short-term and long-term thinking in software development. It argues that prioritizing short-term gains can lead to accumulating technical debt, which ultimately hinders long-term productivity and innovation. The author emphasizes the importance of aligning immediate actions with long-term goals to ensure sustainable development practices.

In this article, we present our vision of architecture for intelligent security system to be used as reference point for the description of our security system implementation. We analyze the functions, structure and knowledge and data storing and processing aspects of the product as a reference architecture.

Formulation of the goals of the article

The purpose of this article is to explore and substantiate the concept of implementing an intelligent security system that utilizes artificial intelligence to enhance risk management efficiency, predict incidents, and ensure situational awareness in real-time.

Tasks of the Article

1. To describe the concept of a situation-oriented security system that detects and analyzes the development of situations in real-time.
2. To examine the use of GFO ontology for creating a general specification of concepts and objects within the system.
3. To investigate the implementation of access control management and video surveillance functions within the system.
4. To assess the role of the knowledge base in utilizing typical incident scenarios and handling protocols to enhance the system’s efficiency.
5. To demonstrate the advantages of implementing artificial intelligence in the security system and its impact on reducing risks and increasing productivity.

Presentation of the main material

Information system requirements

The implementation of an intelligent security system in residential community necessitates a sophisticated and resilient infrastructure capable of managing numerous access points, communal facilities, and large physical environments. This encompasses the integration of diverse security systems, including surveillance cameras, access control mechanisms, and alarm systems, distributed across various edifices and shared spaces.

The analysis of literature [24-27] allows us to specify several functional domains in security system including Access Control, Surveillance and Monitoring, Cybersecurity, Emergency response management, Environmental monitoring, Maintenance and upgrade (fig. 1).

The functional domains are further subdivided into specific functions. For example, Access Control includes Authentication and Authorization, Visitor Management, Gate and Entrance security, Shared space access. Surveillance monitoring has Video Surveillance, Intrusion Detection, Remote monitoring, Data storage and management.

In our first iteration of product, we are focusing on the implementation of Access control and Surveillance and Monitoring subsystems.

Let’s specify the main principles and assumptions for the proposed intelligent security system:

- The intelligent security product is designed as a situation and context aware system, due to the necessity for timely identification of security threats and producing the corresponding responsive actions.
- The system will leverage experiential knowledge derived from historical contexts to inform decision-making processes, thereby enabling proactive measures and facilitating reasoning regarding potential developments in the current scenario.
- The knowledge base will undergo continuous refinement and integration through the utilization of feedback data sourced from various sensors.

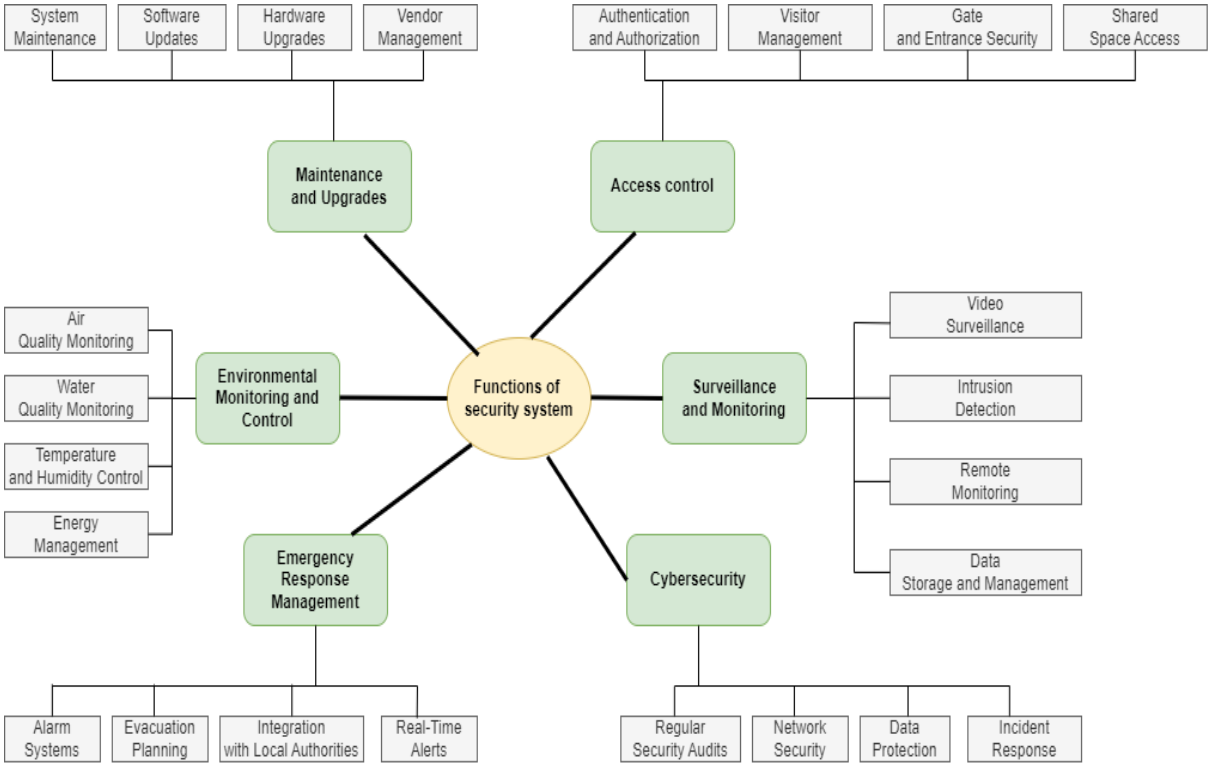

Figure 1. The functions of intelligent security system

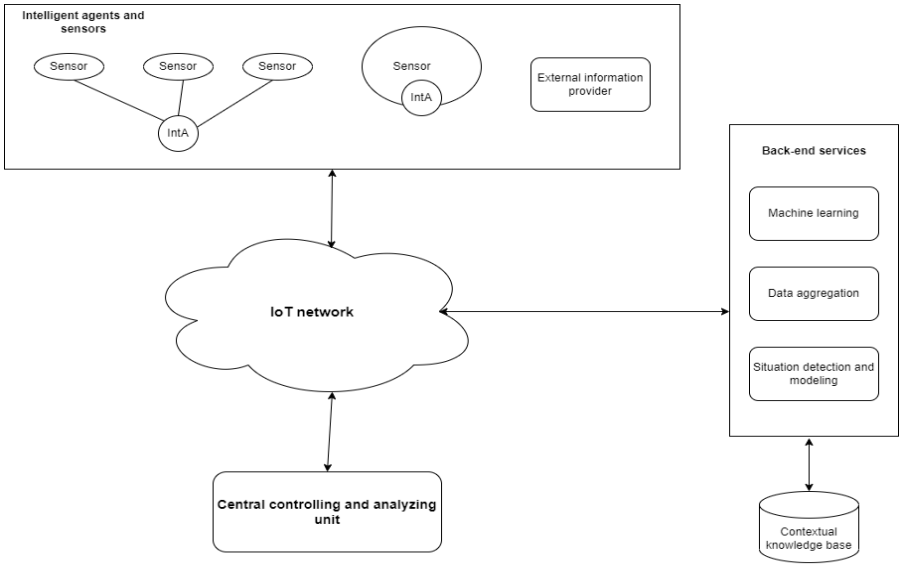

Figure 2. The structure of intelligent security system

- The system will facilitate both localized analysis and decision-making, as well as overarching analytical assessments, thereby coordinating the interactions between its components.
- The architectural solutions will account for the computational resource demands associated with diverse tasks, employing a combination of local, edge, and cloud computing.

The proposed structure of intelligent security system (fig. 2) has such parts:

- Intelligent agents – task-oriented, autonomous units, often integrated with intelligent sensors. Agents can be implemented either locally on specific sensor device or serve a group of devices. Intelligent agents can be also functionally specialized. For example, on performing object recognition or internet search for information.
- Back-end is represented by software services are often implemented as cloud services. They perform resource intensive computations and are built according to SOA requirements.
- Front-end or Central controlling unit which forms and analyzes the overall picture of security situation, using the information supplied by sensors and services and knowledge from the knowledge base and interacting with security staff members.

To maintain the uniformity of conceptual meanings within an intelligent system, such a system ought to be founded upon a shared formal conceptualization (ontology). This ontology must be capable of accommodating both temporal and spatial conceptualizations, thereby constituting a 4-dimensional ontology.

For the purposes of this research, the General Formal Ontology (GFO) has been selected [28]. GFO serves as a foundational 4-dimensional ontology that offers a coherent framework for the conceptualization of forms, modes, and objects across diverse levels of abstraction and granularity. It incorporates methodologies derived from mathematical logic, philosophical inquiry, artificial intelligence, and linguistic analysis.

GFO is using the concepts of topoids, chronoids, configuroids to model spatial, temporal and structural aspects of reality. Situoids and situations are used to represent contexts.

We model the intelligent security system as situation-aware system, able to detect, recognize the important situations, make decisions in them and take appropriate actions. Thus, situoids *Su* and its time slices – situations *Sit* - become the central elements of modeling in proposed system.

Situoids exhibit the following characteristics:

- Contextual Complex. Situoids are used to encapsulate the entirety of a context or situation, encompassing all pertinent entities along with their interrelations.
- Temporal and Spatial Boundaries. A situoid is demarcated both temporally and spatially, indicating that it exists within a defined time frame and a specific spatial domain.
- Dynamic Characteristics. Situoids possess the capacity to undergo transformations over time as the entities and their interrelationships develop within the context.

A situoid Su can be specified through its goal Gl and regarded as a transition between two bounding situations $(Sit_{st}^{su}, Sit_{end}^{su})$, namely the starting state and the intended target state. Multiple intermediary situations are implied to exist between those bounding states. Each situoid is associated with a topoid, a chronoid, and a configuroid, signifying that it happens within both spatial and temporal dimensions, while the world experiences structural modifications during its duration. Situations are situoid states in specific time moments. We can consider them as slices of situoids in the time points. Each situation, like a Situoid can be considered as a whole, but also analyzed structurally.

The intelligent security system operation is organized around processing the interrelated conceptual knowledge models. There are several types of models. Environment models Cm_{env} are built based on the objects recognized in the environment and store the parameters of those objects supplied by sensors or external data sources. Contextual/task models Cm_{con} are keeping data relevant to specific task, situation, goal. They are formed as a subset $Cm_{env,tc} \supseteq Cm'_{env,tc}$ of the Environment model for current time t_c. with additional objects relevant to the intention $Gl_{int,tc}$, provided by corresponding pattern from knowledge base.

$$Cm_{con,tc} = (Cm'_{env,tc}, Gl_{int,tc}, t_c) \quad (1)$$

Intelligent agents monitor object parameters based on the information from a sensor of a group of sensors interpreted as parameters of objects from the ontology On. They maintain the local Environment model and share it with the Central Unit. The operation and decision-making of agents is dependent on the set of contextual models, describing the actions to take in various contexts. Those models are supplied and periodically updated by the system's back-end services which are running learning and pattern-matching algorithms.

Back-end service collects information from intelligent agents and creates and maintains its own global Environment model $Cm_{env,gb}$. This model is used to detect and anticipate situations, coordinate the actions of agents, make system-wide decisions, and present information to human personnel working with the Front-end unit.

The system uses experiential knowledge for detecting situations, planning and anticipating the situation's development. This knowledge is stored contextually, that is a key for retrieval is context similarity. When a system looks for information in a knowledge base it looks for the knowledge about the similar task in similar environment configuration. Or, in case of detecting situations it looks of possible situations/threats which can happen in contexts, like current. When similarity is established, the system makes a mapping between situation and knowledge pattern. In this way access to knowledge represented by patterns is provided.

Therefore, a function F_{sim}, measuring the distance between the current context (1) and the key-context $(\mathrm{Cm}_{\mathrm{env}}^{\mathrm{kb}}, \mathrm{Gl}_{\mathrm{int}}^{\mathrm{kb}})$ in knowledge base should be implemented. In the process of searching the value of this function should be minimized:

$$\mathrm{F_{sim}}: \left(\left(\mathrm{Cm}'_{\mathrm{env,tc}}, \mathrm{Gl}_{\mathrm{int,tc}} \right), \left(\mathrm{Cm}_{\mathrm{env}}^{\mathrm{kb}}, \mathrm{Gl}_{\mathrm{int}}^{\mathrm{kb}} \right) \right) \rightarrow \min \quad (2)$$

Back-end services access, maintain and update contextual knowledge base. They develop, maintain and update policies and models used by intelligent agents and sensors.

Situation detection is done by a service monitoring the current Environment model $Cm_{env,tc}$ on cues and patterns related to situations, which could happen in current context $Cm_{con,tc}$. For this it monitors the set of cues. Each cue is a condition (pattern) with weight, reflecting its importance.

$$Cue = (Cm_{cue}, w_{cue}) \quad (3)$$

Cues are ordered according to their weight. Cues leading to the situations with greater impact have more weight. The impact is deduced from the knowledge base. If an important cue is detected, Central unit updates the data collection policy, allowing it to confirm/decline the presence of a situation.

The intelligent security system uses situoids from GFO to model the dynamics of situations development. This development is presented as a time-ordered sequence of situations $(Sit_{t1}, Sit_{t2}, .., Sit_{tk})$ with the corresponding sequence of configurations $(Cf_{t1}, Cf_{t2}, \dots, Cf_{tk})$. Each configuration in sequence is represented as knowledge graph:

$$Cf_{ti} = (SV_{con}, SE_{rel}, t_i) \quad (4)$$

Where $SV_{con}$is a set of nodes, corresponding to objects and SE_{rel} is a set of relationships used in situation specification. Both objects and relationships are classified according to the system's ontology. Specific configurations in the sequence of configurations can be different, which reflects the situation configurations dynamics in the process of task execution. The transitions between situations are modeled using experiential knowledge as structures of actions or events which cause the transition.

The intelligent security system for a large residential community is a complex system, incorporating many sophisticated, novel and un-tested features. According to Gall's law [29] such system cannot be built from scratch, but should be the result of evolution of simpler, practically viable systems. This approach is supported by Agile software development methodology [30] and the idea of Minimum Viable Product, stating that a version of product should include just enough features to be usable for early customers [31].

At the same time the developers should be lose track of the initial vision, having it as a long-term strategy of product development and updating it considering the results of practical implementation and customer feedback on all stages of product growth. In this part of the article, we present the initial implementation of the security system and show how it is mapped to the overall vision of the intelligent security.

Representing the knowledge base as a set of scenarios

For the first iteration of intelligent security system, we decided to represent the knowledge as a set of typical security scenarios. Each scenario is a situoid, having an initial situation with specified triggering conditions, the sequence of intermediary situations, describing the actions and possible situation trajectories and also the final situation, completing the situoid. In Table 1 we present as an example the 20 selected scenarios and their descriptions.

The initial implementation is limited to Access control and Video Surveillance subsystems, which includes the following objects types:

1. Access Control Subsystem
   - Access Points: Gates, doors, or barriers equipped with control mechanisms.
   - Access Cards/Badges: Physical tokens (e.g., RFID cards) used by residents and staff.
   - Biometric Devices: Fingerprint or face recognition devices used to identify and authenticate individuals.
   - Control Panel: Central unit for managing access control, storing credentials, and logs.
   - Residents/Guests: Individuals who are authorized (or seek authorization) to enter the community.
   - Guards/Staff: Security personnel responsible for monitoring and decision-making.
   - Emergency Response: Mechanisms for handling alarms or security breaches.
2. Video Surveillance Subsystem.
   - Cameras: Positioned at strategic locations to monitor and record video.
   - DVR/NVR: Digital/Network Video Recorder for storing video footage.
   - Motion Sensors: Devices that trigger recording or alerts based on detected movement.
   - Video Management Software (VMS): Software that displays, processes, and analyzes video feeds.
   - Monitors: Screens for security staff to observe live video feeds.
   - Analytics Engine: AI-based system for object detection, face recognition, and anomaly detection in video footage.

T**able 1**

Typical security scenarios

#	**Scenario name**	**Scenario description**
1	Resident Entry Using Access Card	A resident swipes their card at the gate, which is recognized by the control panel, and the gate opens. The camera records the event
2	Guest Entry with Guard Approval	A guest arrives, speaks with a guard at the entry, and the guard verifies the guest's identity via video and grants access.
3	Suspicious Motion Detection at Night	A motion sensor detects movement near the perimeter after dark. Cameras zoom in and record. Guards are alerted.
4	Delivery Personnel Access with Time-Limited Code	Delivery personnel use a one-time access code at a side entrance. The code is validated by the control panel, and video records the entry.
5	Emergency Evacuation Activation	A fire alarm triggers automatic unlocking of all gates and doors for evacuation. Cameras capture the evacuation process.
6	Denying Entry Due to Invalid Credentials	An individual uses an expired access card. The control panel denies access and records the event in the system. The camera captures the attempt
7	Unauthorized Entry Attempt Detected by Video Analytics	Video analytics detect a person attempting to climb over a gate. The system triggers an alarm, and security staff are alerted.
8	Remote Video Monitoring by Guards	Security personnel remotely monitor multiple camera feeds from the control room. They manually intervene when they see suspicious behavior.
9	Biometric Identification of a Resident	A resident uses a fingerprint scanner to gain entry. The system verifies the print and grants access while the camera records the event.
10	Late-Night Patrols with Motion Detection	Guards patrol the area, and their movement triggers video recording at specific points. Surveillance footage is analyzed for anomalies

11	Car Entering via License Plate Recognition	A resident's car enters the community, and the system uses license plate recognition to automatically open the gate. Cameras track the vehicle
12	Nighttime Surveillance in Low Visibility	Infrared cameras capture footage in low light. Video is stored, and guards are alerted if unusual movement is detected by motion sensors.
13	Scheduled Access for Maintenance Staff	Maintenance workers are scheduled to work, and their access cards are valid only for a specific time. Cameras record their activity for accountability
14	Power Failure and Backup Operation	In case of a power outage, the system switches to backup power. Key access points remain functional, and critical cameras continue recording.
15	Remote Access Monitoring by Residents	A resident uses a mobile app to view live footage of their home surroundings, as captured by the security cameras
16	Guard-Assisted Access for VIP Visitors	A VIP guest arrives, and the guard personally escorts them, granting access via the control panel while recording their movement via cameras
17	System Health Check	The system automatically performs a health check, verifying that all cameras, sensors, and access points are operational. Alerts are sent for any issues
18	Tailgating Detection	Video analytics detect two individuals entering after one authorized person swipes their card. An alert is sent to guards for manual verification
19	Camera Failure Detected and Reported	The system detects that a camera is no longer recording or has been tampered with. The control center is notified, and maintenance is dispatched.
20	Perimeter Breach Alert	The fence perimeter sensor detects a breach, and nearby cameras automatically pan and zoom to capture the event. The system alerts guards and records the footage

Specific subsystems and objects cooperate across scenarios. Thus, access control and video surveillance systems cooperate by sharing event triggers. For instance, an unauthorized entry attempt in the access control system will immediately activate cameras to record. Guards and security personnel continuously monitor both access control logs and video footage. In critical events, they can override automatic systems. Cameras are often triggered by access points when a person swipes their card or uses biometric identification, ensuring that both access and corresponding video are logged for cross-verification. Video analytics provide an additional layer of security by detecting anomalies (e.g., tailgating, suspicious activity) and linking with the access control system to alert guards or deny access.

The structure of implemented security system includes sensors (for access control and surveillance cameras), back-end and front-end parts.

Back-end architecture solution

For the development of the backend part of the Intelligent security system, the powerful Django framework was chosen, which is famous for its ability to provide rapid development of reliable and scalable web applications. Django, being a high-level Python web framework, is based on the "Model-View-Template" (MVT) architecture, which is a variation of the classic "Model-View-Controller" (MVC) pattern. Django Rest Framework (DRF) provides a user-friendly and flexible interface for query processing and data management, making it an ideal choice for backend web application development.

The main components of this architecture are:

1. Models

Models in Django are responsible for managing data and defining the structure of the database. They represent application entities, such as users, products, or situations, as Python classes. Each model class defines a set of fields that correspond to columns in the database.

The model uses Django's ORM (Object-Relational Mapping) to interact with the database. The ORM allows you to automatically convert data from a database to Python objects and vice versa, which greatly simplifies the work with the database.

2.Views

Views are responsible for processing HTTP requests that come from users. In the traditional MVC pattern, views correspond to controllers. In Django, views accept requests, interact with models, execute the necessary business logic, and return a response in the form of an HTTP response.

Views can return different types of responses, such as HTML pages, JSON, or even redirects to other URLs. Django also supports class-based views, which allow you to organize your code more modularly and reuse common functionality.

3. Templates

Templates are responsible for rendering the final HTML that the user receives. They contain static HTML code as well as dynamic elements such as variables and template tags used to display data from models. Django provides a built-in template engine that allows to create reusable interface components. Templates can be nested, allowing to create complex interfaces using basic patterns for common elements such as headings or navigation menus.

4. URL-manager

The URL manager in Django is responsible for routing requests to the appropriate views. It works on the principle of regular expression matching, which allows to define which URLs should handle which kinds. When a

request arrives in a Django application, the URL manager checks its compliance with the set rules and passes the processing to the appropriate view.

5. Middleware

Middleware are intermediate layers that handle requests and responses between the server and views. They can modify or process data at different stages of the query lifecycle. Django supports various types of middleware, such as authorization, caching, session processing, and many others.

6. Database

Django supports working with various types of databases, including PostgreSQL, MySQL, SQLite, and others. Using an ORM allows you to abstract from specific SQL queries and work with the database through a user-friendly Python interface. This greatly simplifies data migration and support for cross-platform applications.

7. Migrations

Django provides a powerful tool for managing changes to the database schema — migrations. Whenever the model changes, Django automatically creates a migration that reflects those changes in the database. This makes it easy to track and control all changes in the data structure.

A PostgreSQL database was used to store the information.

PostgreSQL is a powerful, open-source object-relational database management system (DBMS). It is highly reliable, scalable, and flexible, making it a popular choice for both small applications and large enterprise systems. Key Features of PostgreSQL:

1.Support for ACID transactions: PostgreSQL ensures full compliance with the principles of ACID (Atomicity, Consistency, Isolation, Durability), which guarantees the reliability of data processing.

2.Extensibility: PostgreSQL allows you to add new features, such as data types, operators, and indexes, without changing the source code of the database.

3.Support for complex queries: PostgreSQL has a powerful query processing engine, including support for subqueries, joins, aggregate functions, and multidimensional arrays.

4.Implementation of a variety of data types: In addition to the usual types, PostgreSQL supports JSON, XML, hstore (key-value), geometric types, and also allows you to create your own data types.

5.Scalability: PostgreSQL scales well both vertically (on a single machine) and horizontally (across multiple machines), providing high performance as the volume of data grows.

6.Advanced Concurrency Support: PostgreSQL supports parallel query processing, which can reduce the execution time of complex queries on large amounts of data.

7.Security: PostgreSQL provides the ability to configure authentication, encryption, and data access control at various levels.

8.Replication and Disaster Recovery: PostgreSQL supports data replication for high availability and recoverable backup.

To ensure a stable and secure deployment environment, the Debian operating system and the Nginx web server are used. Debian provides a stable platform for running server components, while Nginx provides efficient HTTP request service, load balancing, and caching.

The project uses RQ Worker (Redis Queue Worker) to handle background tasks. It is a tool for performing asynchronous tasks that works in tandem with Redis, a database that functions as a message queue.

For interaction between the backend and intercom systems, particularly BasIP, the requests library and the BasIP API are used, which provides access to various functions of intercom systems via HTTP requests. (obtaining information about the status of devices, call control, configuration of parameters, etc.)

The back end for video surveillance system (fig. 3) solves the following tasks:

- Proxy Streams
- Record streams (optional)
- Provide links to the archive and Live footage

It includes the following components. The main components that are used to implement the tasks are:

- ZoneMinder system (archiving)
- nginx with rtmp module (issuing a link to live streams)
- nginx from ngx_http_mp4_module (issuing a link to the archive)
- API (interaction between video surveillance system components and other systems)
- ffmpeg та ffprobe

To isolate processes and simplify the deployment of the video surveillance solution, the capabilities of the docker platform were used: a Dockerfile with instructions was generated, with the help of which an image containing the necessary packages for nginx to work with rtmp and ngx_http_mp4_module modules was assembled. Based on the assembled image, the container is started with port forwarding and mounting partitions in which nginx configuration files and corresponding modules, logs, as well as a folder with records are dropped. This makes it possible to make changes from the host operating system without having to rebuild the image.

For operations with streams and archives, an application programming interface (API) is implemented. The main points that are embedded in the API:

- Adding, updating, deleting incoming streams
- Viewing information on streams that are already being processed by the system
- Archiving operations

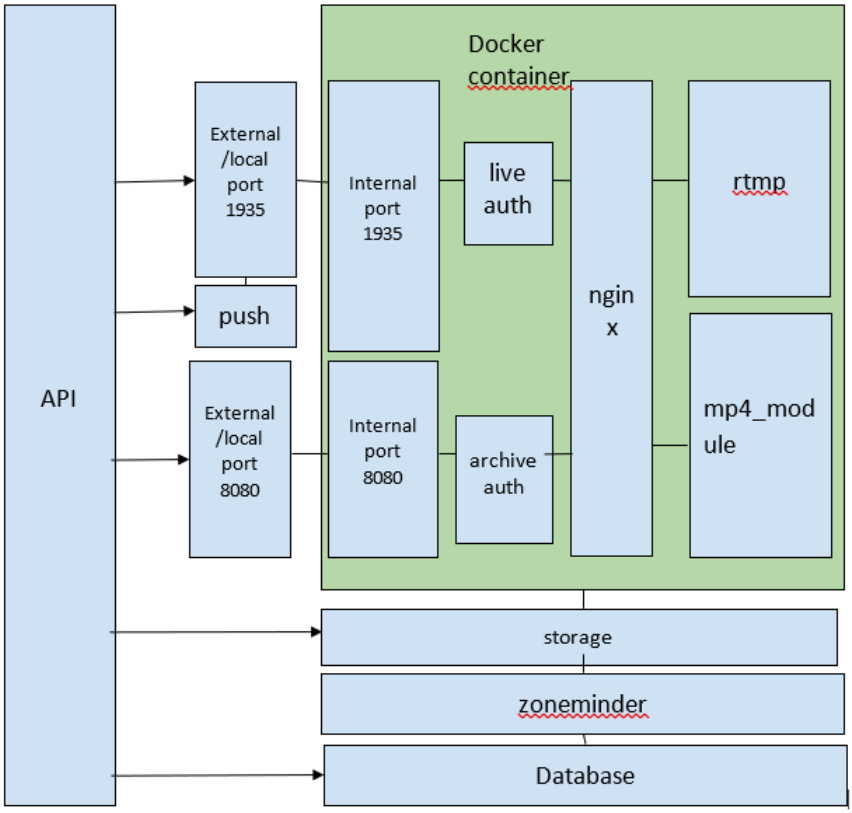

Figure 3. The architecture of back-end video surveillance sub-system

The new input stream is added as a result of such operation sequence:

- A request is sent to the corresponding API point, in which the name and URL of the incoming stream are passed.
- The API creates a service at the core system level, the task of which is to redirect the input stream (push) to the nginx rtmp module that is running in the docker. Push is implemented using the open source ffmpeg tool. Before launching the service, the metadata of the stream is determined using the ffprobe tool. Metadata provides insight into the structure of the input stream and allows you to determine whether the audio needs to be transcoded to supported codecs. The primary audio codec is defined as aac. H264 is adopted as a video codec.
- After working out the enpoint for adding the input stream, the URL from the rtmp module is returned to the user.

When adding a stream to a recording, you have the option to select the recording function (record continuously or motion-only).

The input stream can be added to the proxy. Then the ability to play via the rtmp protocol will become available. Also, the input stream can be added to the record (added via the rtsp protocol). If there is a need to add a stream both for proxy and for writing at the same time, the input stream on zoneminder can be rtmp from the output of the rtmp module. That is, you do not need to receive the stream from the camera or panel twice.

Also, the API provides the ability to set the recording depth in days. At the same time, a corresponding rule (filter) is created at the level of the zoneminder system, which is executed periodically.

The database of the zoneminder system is separated from the zoneminder to provide flexibility in the creation of backups of this database. Zoneminder, along with the API and docker container, must be on the same physical server that has enough resources to handle a large number of threads. At the level of the operating system, as well as systems and services launched on the basis of the docker platform, tuning must be carried out to ensure optimal operation of the server under high loads.

Front-end implementation

For the development of the front-end part of the Smart LCD product, the popular React.js library was chosen, and the code is written in TypeScript, which, being a strictly typed JavaScript extension, improves the quality of the code and reduces the number of errors during development. When combined with React, TypeScript contributes to the creation of more robust and scalable applications.

React, as a library for building user interfaces, provides the reuse of components and the virtual DOM, which improves application performance. Integration with TypeScript makes these applications even more flexible and reliable.

For the rapid development and assembly of the project, Vite, a state-of-the-art assembler that provides fast build times and real-time module updates, is used. Thanks to Vite, the development process becomes even more efficient, and the results are displayed faster in the browser.

The Smart LCD project uses the capabilities of the React Router DOM library to implement single-page application-style routing (SPA). This ensures smooth and seamless navigation between different sections of the application without the need to reload pages.

Thanks to the React Router DOM, the project implements:

- Dynamic Routes – Easily defined and manipulated routes with parameters.
- Continuous Navigation: Navigating between pages is quick and without reloading, resulting in a more interactive user experience.
- Nested routes – Support for nested routes allows to create complex navigation structures.
- Route protection: Implement mechanisms to protect access to specific pages.

To manage the state of data in React applications, TanStack Query is used, which allows you to efficiently cache and synchronize data, which significantly improves the API experience.

To make HTTP requests to the server, Axios component is used, which provides a flexible interface for interacting with the backend. To store state in the front-end part, the Zustand library is utilized. It provides an easy and flexible approach to state management in React applications, which simplifies the architecture and improves application performance. Ant Design is used in the project as a popular component library for React, providing high-quality, ready-to-use UI components. It allows to create a modern, minimalist design with the ability to easily customize themes, and it also integrates with other tools to create efficient and elegant web applications. Using Ant Design helps ensure a consistent user experience and significantly speeds up the development process.

Effective interaction between the backend and the frontend is critical for successful web application development. Using the Django Rest Framework (DRF) to create a RESTful API makes it easy to handle requests from the frontend, while React combined with TanStack Query and Axios ensures seamless API integration.

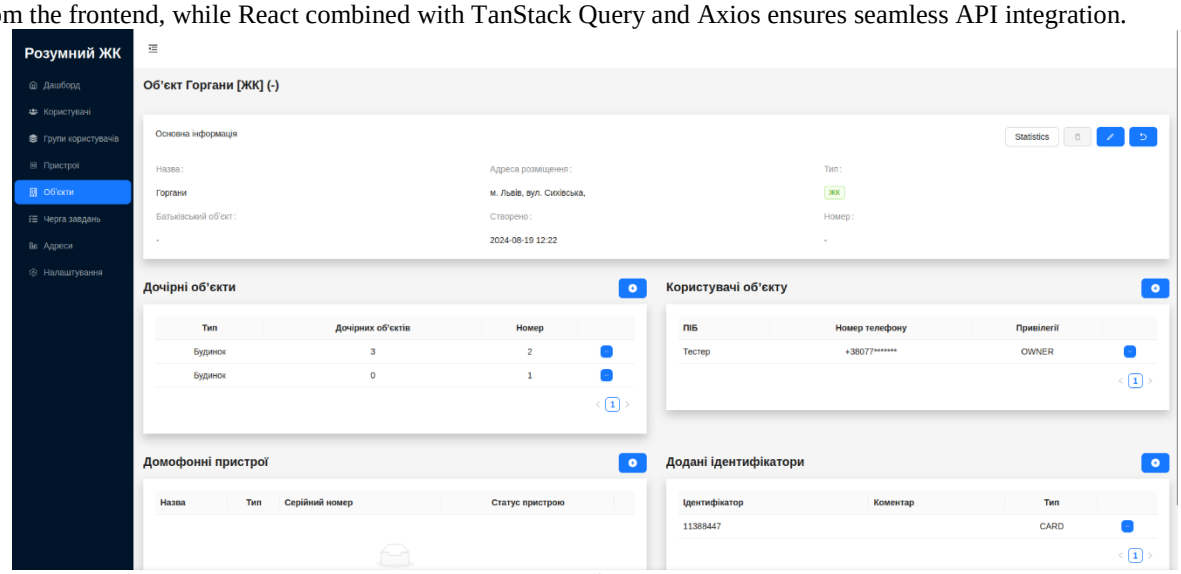

Figure 4. The list of registered objects

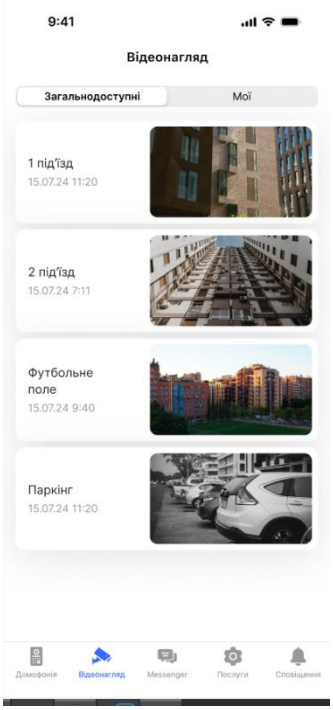

Figure 5. Mobile surveillance application screen

The figure 4 presents the list of registered objects, with corresponding properties in application management interface. The figure 5 – one of the screens in mobile surveillance application allowing to see all observed locations.

Conclusions from this research and prospects for further research in this direction

The article describes the vision of intelligent security system and the first version of its implementation. The system uses a GFO-derived ontology to create the common sets of objects and situational analysis to identify and track security threats. The intelligent security structure implementation includes front-end and back-end parts. The first version of system focuses on the implementation of basic functionality for access control and surveillance, allowing to enhance the system by adding more intelligent features.

Introducing artificial intelligence (AI) features to the security system can significantly enhance the efficiency, scalability, and intelligence of both access control and video surveillance systems. AI would allow the system to learn from past incidents, recognize patterns, and make decisions autonomously, improving response times and reducing false positives.

Using AI we can enhance the functioning of security system and its objects.

For example, in access control subsystem we can introduce:

- AI-Powered Access Manager: An intelligent agent that continuously analyzes access patterns, detects anomalies (e.g., unusual times of entry), and flags potential security breaches based on learned behavior.
- AI Identity Verification: Machine learning algorithms that enhance biometric identification (fingerprint, face recognition) by adapting to changes in a person's physical characteristics over time, thus reducing error rates.
- Behavioral Profiling Agent: This AI agent builds behavioral profiles of residents, staff, and frequent visitors, allowing the system to preemptively detect suspicious deviations from normal behavior (e.g., unusual time of access or multiple failed entry attempts).

In video surveillance subsystem following ai-based enhancements are planned:

- AI-Based Video Analytics: Machine learning models for pattern recognition, object detection, and anomaly detection (e.g., identifying abandoned objects, unusual movement patterns, or unauthorized entry through unguarded areas).
- Facial Recognition with Deep Learning: Advanced facial recognition that not only identifies known individuals but also detects stress levels, emotional states, or intent based on facial cues.
- Predictive Surveillance: An AI module that processes real-time video feeds, predicts suspicious activity (e.g., loitering near critical points) based on historical data, and alerts guards before incidents occur.

Back-end can be enhanced by adding the learning functionality:

- Machine Learning Model Manager: A backend system that continually refines AI models using past access data and surveillance footage. This system provides automated suggestions for system improvements (e.g., better camera placement, more restrictive access rules).
- Automated Incident Learning: A learning module that reviews and categorizes past incidents (security breaches, false alarms, normal behavior) to train AI models, improving accuracy over time.
- Context-Aware System: AI that analyzes contextual data such as weather, time of day, and public events to adjust security measures accordingly (e.g., increase surveillance during festivals or modify access control rules during extreme weather events).

The introduction of intelligent features in the product will be done incrementally, using the customer feedback to update our vision of intelligent security system.

References

1. Iqbal, H., Sarker., Iqbal, H., Sarker., Hasan, Furhad., Raza, Nowrozy. (2021). 1. AI-Driven Cybersecurity: An Overview, Security Intelligence Modeling and Research Directions. doi: 10.1007/S42979-021-00557-0

2. R. Raimundo and A. Rosário, "The impact of artificial intelligence on data system security: A literature review," Sensors, vol. 21, no. 21, p. 7029, 2021, doi: 10.3390/s21217029

3. L. Cui, G. Xie, Y. Qu, L. Gao, and Y. Yang, "Security and privacy in smart cities: Challenges and opportunities," IEEE access, vol. 6, pp. 46134–46145, 2018, doi: 10.1109/ACCESS.2018.2853985.

4. H. Golnabi, "Smart sensors development and applications." 2023. doi: 10.1615/faim1998.570.

5. Sharma et al., "Recent trends in AI-based intelligent sensing," Electronics, 2022, doi: 10.3390/electronics11101661

6. Anghel and T. Cioara, "Emerging sensors techniques and technologies for intelligent environments," Sensors, 2022, doi: 10.3390/s22176427

7. Castelfranchi, "Intentions in the Light of Goals," Topoi, vol. 33, no. 1, pp. 103–116, Apr. 2014, doi: 10.1007/s11245-013-9218-3.

8. X. Kuangdi, "Deployment and operation," Advances in information security, 2023, doi: 10.1007/978-3-031-29269-9_14.

9. Kott, "Autonomous intelligent cyber-defense agent: Introduction and overview," arXiv.org, 2023, doi: 10.48550/arXiv.2304.12408

10. Adnan, Ramakić., Zlatko, Bundalo., Dusanka, Bundalo. (2024). An Example of Intelligent Security System Based On Deep Learning. Journal of Circuits, Systems, and Computers, doi: 10.1142/s0218126624502086

11. Rahim, Y. Zhong, T. Ahmad, S. Ahmad, P. Pławiak, and M. Hammad, "Enhancing smart home security: anomaly detection and face recognition in smart home IoT devices using logit-boosted CNN models," Sensors, vol. 23, no. 15, p. 6979, 2023, doi: 10.3390/s23156979

12. J. Albusac, D. Vallejo, L. Jimenez-Linares, J. J. Castro-Schez, and L. Rodriguez-Benitez, "Intelligent surveillance based on normality analysis to detect abnormal behaviors," International Journal of Pattern Recognition and Artificial Intelligence, vol. 23, no. 07, pp. 1223–1244, 2009, doi: 10.1142/S0218001409007612.

13. O. Dolgov., B. Safoklov., D. Shavelkin. "Intelligent Access Control Systems", (2022). doi: 10.1109/ICIEAM54945.2022.9787125

14. E. Schiller. "IoT-Based Access Management Supported by AI and Blockchains". Electronics, (2022), doi: 10.3390/electronics11182971

15. H. F. Atlam, A. Alenezi, R. J. Walters, G. B. Wills, and J. Daniel, "Developing an adaptive Risk-based access control model for the Internet of Things," presented at the 2017 IEEE international conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and ieee smart data (SmartData), IEEE, 2017, pp. 655–661.

16. S. Dutta, S. S. L. Chukkapalli, M. Sulgekar, S. Krithivasan, P. K. Das, and A. Joshi, "Context sensitive access control in smart home environments," presented at the 2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), IEEE, 2020, pp. 35–41

17. Teich., N. Högasten., T. Hoelter, K. Strandemar. "Smart surveillance camera systems and methods"., 2014.

18. Marman, S. Mahesh. "Intelligent high resolution video system." U.S. Patent 9,584,710, issued February 28, 2017.

19. S. Nikouei., Y. Chen., S. Song., R. Xu., C. Baek-Young, T. Faughnan. "Smart Surveillance as an Edge Network Service: from Harr-Cascade, SVM to a Lightweight CNN". arXiv: Computer Vision and Pattern Recognition. 2018.

20. J., Castro., M. Delgado., J. Miguel, D. Ruiz-Lozano. "Intelligent surveillance system with integration of heterogeneous information for intrusion detection". Expert Systems With Applications, 2011 doi: 10.1016/J.ESWA.2011.02.165

21. J. Fernández et al., "An intelligent surveillance platform for large metropolitan areas with dense sensor deployment," Sensors, vol. 13, no. 6, pp. 7414–7442, 2013, doi: 10.3390/s130607414.

22. Kelly, "Determining factors that affect long-term evolution in scientific application software," Journal of Systems and Software, vol. 82, no. 5, pp. 851–861, 2009, doi: 10.1016/j.jss.2008.11.846.

23. V. Khorikov, "Short-term vs long-term perspective in software development," Enterprise Craftsmanship. Accessed: Sep. 19, 2024. [Online]. Available: https://enterprisecraftsmanship.com/posts/short-term-vs-long-term-perspective/

24. https://www.larnitech.com/news/security-systems-in-a-smart-house/

25. M. Wang and C. Han, "The design of intelligent residence property management information system (IPMIS) based on e-business," presented at the The CRIOCM 2006 International Symposium on "Advancement of Construction Management and Real Estate, 2006, pp. 1–9.

26. R. Cimorelli Belfiore and A. L. Ferrara, "Security Analysis of Access Control Policies for Smart Homes," presented at the Proceedings of the 28th ACM Symposium on Access Control Models and Technologies, 2023, pp. 99–106.

27. M. Pech, J. Vrchota, and J. Bednář, "Predictive maintenance and intelligent sensors in smart factory," Sensors, vol. 21, no. 4, p. 1470, 2021, doi: 10.3390/s21041470

28. Loebe, Frank, Patryk Burek, and Heinrich Herre. "GFO: The General Formal Ontology." Applied Ontology 17, no. 1 (2022): 71–106. https://doi.org/10.3233/AO-220264.

29. Gall, John. "Systemantics: How Systems Really Work and How They Fail." Quadrangle/New Times Book, 1975.

30. "Agile Manifesto." Agilemanifesto.org.

31. G. Villalobos Rodríguez, M. Vargas Montero, J. Rodriguez Ramirez, and L. A. ARAYA-CASTILLO, "Lean start-up as a strategy for the development and management of dynamic entrepreneurships," Dimensión Empresarial, vol. 16, no. 2, pp. 193–208, 2018