Embedded systems

Embedded systems are self-contained control systems or computer systems designed for particular purpose with bare necessary peripherals needed to run it. In the contemporary world, billions of embedded system devices including telephones, automobiles, electronic appliances and other digital devices are working around us, round the clock.
It contains processor, memory, peripherals, and sensors etc.

While general purpose systems are meant for multiple purposes, embedded systems are designed for specific ones. Embedded systems may not have generic interfaces in most of the cases. Even if they have, it will have a specific purpose such as displaying output, giving input etc. Since they are used for a single purpose, the cost of implementing it will be less. Most of the time, embedded systems are time-critical applications which are not possible in general purpose systems.

Uses and applications

With the advent of electronics and related technology, embedded systems have cast their presence almost everywhere. Embedded systems are widely seen in consumer devices, medical equipment, industrial instruments, transportation systems, military equipment etc.

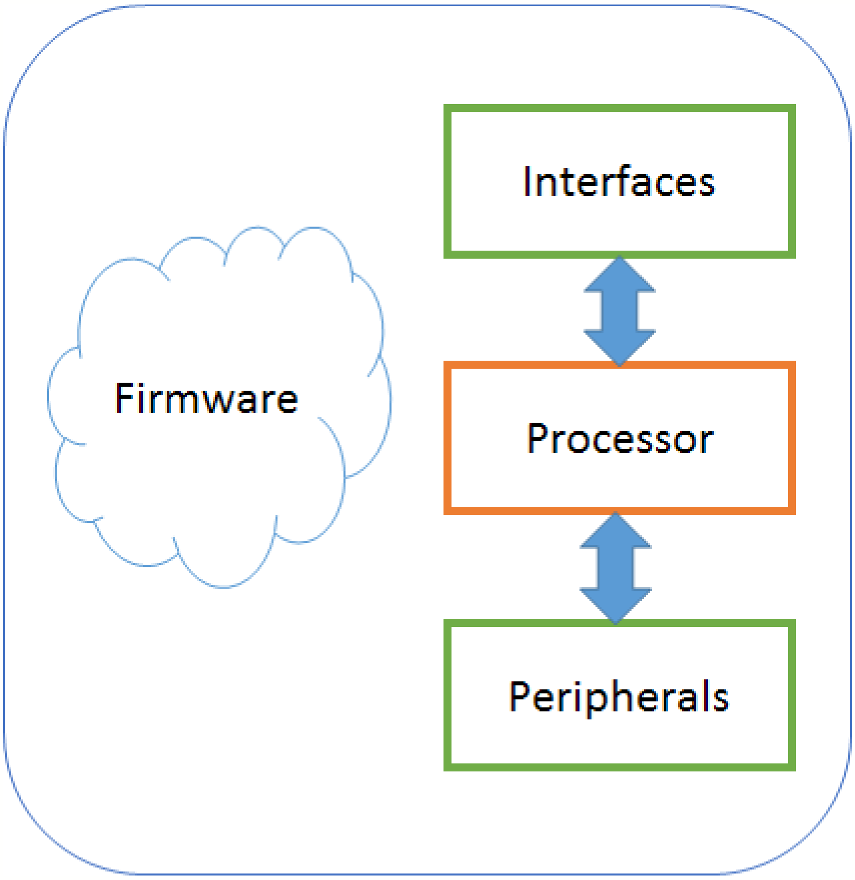
Characteristics

- Embedded systems are designed for specific tasks and are single functioned
- Some of them don't have any external interface at all, some may contain complex interfaces. It's based on the requirements of the system
- Some systems may provide user interfaces directly or through serial or network connection which reduces the overall cost of the system by avoiding displays and input interfaces
- Embedded systems are typically designed to meet real-time constraints
- They are designed with high optimization for memory, power usage, execution time, dimensions, weight, and cost
- Embedded system use a few resources, so developing a system will be a challenging task
- The system should be highly reliable; it is not designed for repairs and cannot be shut down for safety reasons.

Embedded system architecture

Embedded system is a combination of hardware and software. Hardware is designed for the specific purpose and likewise the software. The resources will be very scarce as the application of the system is limited. The hardware and software are interlinked to each other so that the system reduces the burden of cost, time (to bring the product in the market) and tools.
The core part of the embedded system is the processors. Different interfaces and

peripherals are interfaced to the system based on the requirements. The architecture of an embedded system can be simply picturized as below.



An Embedded system – Block diagram

Processor

There are many options while choosing a processor. These include

- General purpose microprocessor
- Microcontroller
- Digital signal processors
- Field programmable gate array (FPGA)
- Complex programmable logic devices (CPLD)
- System on chip (SoC)

Each of these devices has their own advantages and disadvantages; they are used based on the complexity of the application.

Interfaces

Each device will use the interfaces to connect with outside world. There are a bunch of interfaces including serial, parallel, digital and analog interfaces to choose from. As like others, it also depends on the application and particular use. Some of the interfaces available to embedded systems are

- Serial communication interfaces, e.g: RS-232
- Synchronous serial communication interface: e.g: I2C
- USB
- Networks: E.g: WiFi
- Debugging, E.g: JTAG
- Field buses, E.g: CAN

Peripherals

Peripherals are connected to the embedded system to implement specific features. These may include various sensors, displays, input devices, output devices and actuators. Most of the peripherals may need an interface described above.
Some of the peripherals used in embedded systems are

- Displays, E.g: Graphic LCD
- Multimedia cards, E.g: SD Card
- Timers, Counters
- Analog to Digital converters
- Input devices, E.g: keypad
- Output devices, E.g: LED

Software

Another important part of the embedded system is the software. The embedded system software, generally called firmware, defines how the hardware behaves in each circumstance. The system should be capable of taking account of all the available conditions in a system and should predict the output. In some applications, it may need real-time processing of the data, which is called real-time applications and it may need specialized operating systems called real-time operating system.

Designing embedded systems

Embedded system design is an interesting area of work. Each embedded system is designed for a particular application, and it is also a product. So the development of the embedded systems is defined by the embedded development life cycle (EDLC).

While designing embedded systems, we need to consider the hardware and software part of the systems. It involves the same product development pipeline like requirement analysis, market survey and customer feedback, but in later development & implementation and integration stages, one needs to follow the below steps.

- Define system specifications based on the requirement analysis

- Co-design – decide which system should be implemented with hardware and which through software
- Technology selection – select main parts and associated technologies
- Resource allocation – decide the resources needed for the design and testing for this product, this includes budget and people
- Component selection and tools identification
- Hardware design – schematics, layout, PCB manufacturing and board bring-up
- Firmware development and testing
- System integration and testing
- Testing, certifications

Smart city

A smart city is a municipality that uses information and communication technologies (ICT) to increase operational efficiency, share information with the public and improve both the quality of government services and citizen welfare.

While the exact definition varies, the overarching mission of a smart city is to optimize city functions and drive economic growth while improving quality of life for its citizens using smart technology and data analysis. Value is given to the smart city based on what they choose to do with the technology, not just how much technology they may have.

Several major characteristics are used to determine a city's smartness. These characteristics include:

technology-based infrastructure;
environmental initiatives;
high functioning public transportation system;
confident sense of urban planning and
humans to live and work within the city and utilize its resources.

A smart city's success depends on its ability to form a strong relationship between the government -- including its bureaucracy and regulations -- and the private sector. This relationship is necessary because most of the work that is done to create and maintain a digital, data-driven environment occurs outside of the government. Surveillance equipment for busy streets could include sensors from one company, cameras from another and a server from yet another.

Additionally, independent contractors may be hired to analyze the data which is then reported back to the city government. This data could then lead to the incorporation of an application development team that is hired to come up with a solution for the problems found in the analyzed data. This company could become part of the system if the solution requires regular updating and management. Therefore, a smart city's success becomes more focused on building positive relationships than on completing a single project.

Smart city technology

Smart cities use a combination of the internet of things (IoT) devices, software solutions, user interfaces (UI) and communication networks. However, they rely first and foremost on the IoT. The IoT is a network of connected devices -- such as vehicles, sensors or home appliances -- that can communicate and exchange data. Data collected and delivered by the IoT sensors and devices is stored in the cloud or on servers. The connection of these devices and use of data analytics (DA) facilitates the convergence of the physical and digital city elements, thus improving both public and private sector efficiency, enabling economic benefits and improving citizen's lives.

The IoT devices sometimes have processing capabilities called edge computing. Edge computing ensures that only the most important and relevant information is communicated over the communication network.

A firewall security system is also necessary for the protection, monitoring and control of network traffic within a computing system. Firewalls ensure that the data constantly being transmitted within a smart city network is secure by preventing any unauthorized access to the IoT network or city data.

Other smart city technologies include:

application programming interfaces (APIs)

artificial intelligence (AI)

cloud computing

dashboards

machine learning (ML)

machine to machine (M2M)

mesh network

Features of a smart city

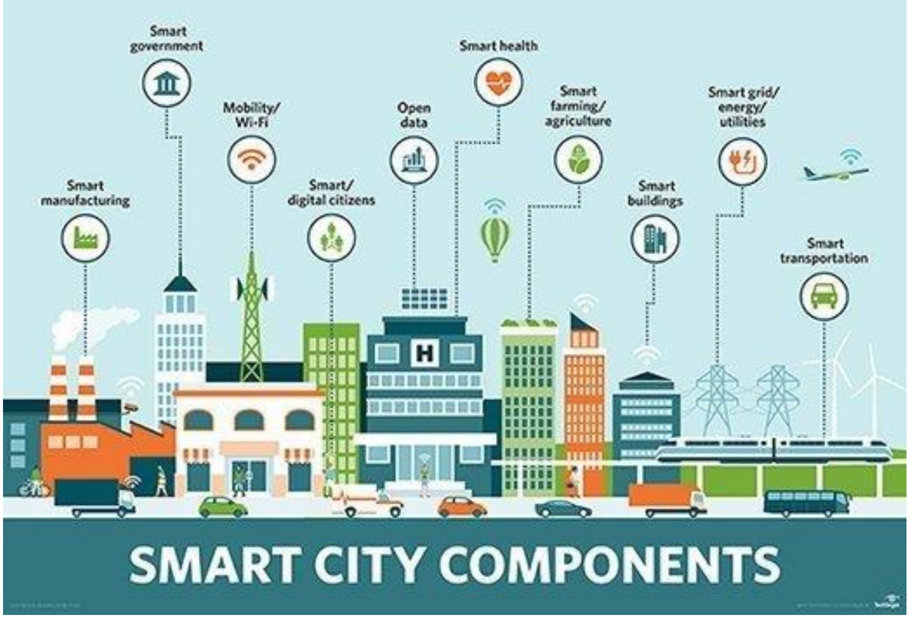
Emerging trends such as automation, machine learning and the IoT are driving smart city adoption.

Theoretically, any area of city management can be incorporated into a smart city initiative. A classic example is the smart parking meter that uses an application to help drivers find available parking spaces without prolonged circling of crowded city blocks. The smart meter also enables digital payment, so there's no risk of coming up short of coins for the meter.

Also in the transportation arena, smart traffic management is used to monitor and analyze traffic flows in order to optimize streetlights and prevent roadways from becoming too congested based on time of day or rush-hour schedules. Smart public transit is another facet of smart cities. Smart transit companies are able to coordinate services and fulfill riders' needs in real time, improving efficiency and rider satisfaction. Ride-sharing and bike-sharing are also common services in a smart city.

Energy conservation and efficiency are major focuses of smart cities. Using smart sensors, smart streetlights dim when there aren't cars or pedestrians on the roadways. Smart grid technology can be used to improve operations, maintenance and planning, and to supply power on demand and monitor energy outages.

Smart city initiatives also aim to monitor and address environmental concerns such as climate change and air pollution. Waste management and sanitation can also be improved with smart technology, be it using internet-connected trash cans and IoT-enabled fleet management systems for waste collection and removal, or using sensors to measure water parameters and guarantee the quality of drinking water at the front end of the system, with proper wastewater removal and drainage at the back end.



Smart city technology is increasingly being used to improve public safety, from monitoring areas of high crime to improving emergency preparedness with sensors. For example, smart sensors can be critical components of an early warning system before droughts, floods, landslides or hurricanes.

Smart buildings are also often part of a smart city project. Legacy infrastructure can be retrofitted and new buildings constructed with sensors to not only provide real time space management and ensure public safety, but also to monitor the structural health of buildings. Sensors can detect wear and tear, and notify officials when repairs are needed. Citizens can help in this matter, notifying officials through a smart city application when repairs are needed in buildings and other public infrastructure, such as potholes. Sensors can also be used to detect leaks in water mains and other pipe systems, helping reduce costs and improve the efficiency of public workers.

Smart city technologies also bring efficiencies to urban manufacturing and urban farming, including job creation, energy efficiency, space management and fresher goods for consumers.

How a smart city works

Smart cities utilize their web of connected IoT devices and other technologies to achieve their goals of improving the quality of life and achieving economic growth. Successful smart cities follow four steps:

Collection - Smart sensors throughout the city gather data in real time.

Analysis - Data collected by the smart sensors is assessed in order to draw meaningful insights.

Communication - The insights that have been found in the analysis phase are communicated with decision makers through strong communication networks.

Action - Cities use the insights pulled from the data to create solutions, optimize operations and asset management and improve the quality of life for residents.

Fostering sustainability with smart cities

Sustainability is another major facet of smart cities. Urbanization is expected to increase even more in the coming years. The United Nations reports that around 55% of the world's population currently resides in an urban area or city; this figure is set to rise 68% throughout the coming decades. Smart technology will help cities sustain growth and improve efficiency for citizen welfare and government efficiency in urban areas in the years to come.

While cities already present environmental advantages, such as smaller geographic footprints that impact fewer ecological systems, they also negatively impact the environment with emissions, such as their extreme usage of fossil fuels. The network of smart city technologies could alleviate these detrimental effects.

Making the switch to an electric public transportation system would not only decrease fuel emissions, but could also pose the advantage of working closely with the city's electric power infrastructure in order to minimize the impact of charging batteries during peak hours of electric use. Furthermore, with proper coordination, electric vehicles could also be used to regulate the frequency of the city's electric grid when they're not in service.

The number of cars used in cities is also expected to decrease as municipalities become smarter. Autonomous vehicles, or self-driving cars, could potentially change a population's perspective on the necessity of owning cars. It is suspected that the adoption of autonomous vehicles will reduce the amount of vehicles owned by civilians, thus decreasing the number of cars on the street and further lowering the emission of detrimental gases.

Smart city challenges and concerns

Smart city initiatives must include the people they aims to help: residents, business people and visitors. City leaders must not only raise awareness of the benefits of the smart city technologies being implemented, but also promote the use of open, democratized data to its citizens. If people know what they are participating in and the benefits it can bring, they are more likely to engage.

Fostering collaboration between the public and private sector and city residents is key to creating a smart citizen who will be engaged and empowered to positively contribute to the city and community. Smart city projects should include plans to make the data transparent and available to citizens, often through an open data portal or mobile app. This enables residents to engage with the data and understand what it is used for. Through a smart city app, residents may also be able to complete personal chores, such as viewing their home's energy consumption, paying bills and finding efficient public transportation.

Smart city opponents worry that city managers will not keep data privacy and security top of mind, fearing the exposure of the data that citizens produce on a daily basis to the risk of hacking or misuse. Additionally, the presence of sensors and cameras may be perceived as an invasion of privacy or government surveillance. To address this, smart city data collected should be anonymized and not be personally identifiable information.

However, perhaps the biggest challenge smart cities face is the problem of connectivity. The thousands or millions of IoT devices scattered across the city would be defunct without a solid connection and the smart city itself would be dead.

Furthermore, public transit, traffic management, public safety, water and waste management, electricity and natural gas supply can be unreliable, especially as a system ages and grows. However, the importance of these operations will only increase as the city expands and the demands on its infrastructure increase. These systems must be constantly maintained and tested to ensure their proper functioning.

Smart cities are also challenged by finding ways to attract and keep residents without a cultural fabric. The cultural essence of an area is oftentimes what attracts residents the most; this is something that cannot be programmed or controlled with a sensor. Therefore, smart cities may falter because they cannot provide a sense of authenticity, distinctiveness or place.

Additionally, smart cities that are being created from the ground up -- like Saudi Arabia's Neom and Arizona's Buckeye which are being built in the desert -- lack an established population and are therefore presented with the obstacle of having to recruit residents. These future smart cities also have no past success to provide confidence. As Neom and Buckeye have been built, concerns have risen over whether or not there is even a sustainable water source available.

Why we need smart cities

The primary goal of a smart city is to create an urban environment that yields a high quality of life to its residents while also generating overall economic growth. Therefore, a major advantage of smart cities is their ability to facilitate an increased delivery of services to citizens with less infrastructure and cost.

As the population within cities continues to grow, it becomes necessary for these urban areas to accommodate the increasing population by making more efficient use of their infrastructure and assets. Smart city applications can enable these improvements, advance city operations and improve the quality of life among residents.

Smart city applications enable cities to find and create new value from their existing infrastructure. The improvements facilitate new revenue streams and operational efficiencies, helping governments and citizens save money.

Examples of smart cities

While many cities across the world have started implementing smart technologies, a few stand out as the furthest ahead in development. These cities include:

Kansas City, Missouri

San Diego, California

Columbus, Ohio

New York City, New York

Toronto, Canada

Singapore

Vienna, Austria

Barcelona, Spain

Tokyo, Japan

Reykjavik, Iceland

London, England

Melbourne, Australia

Dubai, United Arab Emirates

Hong Kong, China

Most of the new smart city projects are concentrated in the Middle East and China, but in 2018, Reykjavik and Toronto were listed alongside Tokyo and Singapore as some of the world's smartest cities.

Often considered the gold standard of smart cities, the city-state of Singapore uses sensors and IoT-enabled cameras to monitor the cleanliness of public spaces, crowd density and the movement of locally registered vehicles. Its smart technologies help companies and residents monitor energy use, waste production and water use in real time. Singapore is also testing autonomous vehicles, including full-size robotic buses, as well as an elderly monitoring system to ensure the health and well-being of its senior citizens.

The smart city initiative of Kansas City, Mo., involves smart streetlights, interactive kiosks and more than 50 blocks of free public Wi-Fi along the city's two-mile streetcar route. Available parking spaces, traffic flow and pedestrian hotspots are all publicly available through the city's data visualization app.

San Diego installed 3,200 smart sensors in early 2017 to optimize traffic and parking and enhance public safety, environmental awareness and overall livability for its residents. Solar-to-electric charging stations are available to empower electric vehicle use, and connected cameras help monitor traffic and pinpoint crime.

In Dubai, United Arab Emirates, smart city technology is used for traffic routing, parking, infrastructure planning and transportation. The city also uses telemedicine and smart healthcare, as well as smart buildings, smart utilities, smart education and smart tourism.

The Barcelona, Spain, smart transportation system and smart bus systems are complemented by smart bus stops that provide free Wi-Fi, USB charging stations and bus schedule updates for riders. A bike-sharing program and smart parking app that includes online payment options are also available. The city also uses sensors to monitor temperature, pollution and noise, as well as monitor humidity and rain levels.

WHAT IS A SMART PARKING SYSTEM? FUNCTIONALITIES AND BENEFITS

If there's one maneuver that stresses drivers, it's looking for a parking space in the city. This, however, is changing with the implementation of smart solutions.

According to Libelium, a company that provides technology solutions based on the Internet of Things (IoT), applying this technology reduces traffic volume by 8%, gas emissions by 40%, kilometers travelled by a car to park by 30% and time spent parking by 43%.

WHAT IS A SMART PARKING SYSTEM?

Thanks to digitalization, smart parking systems are starting to offer **solutions for urban mobility.** This is a system, which, thanks to the Internet of Things and sensor technology, allows real-time data to be obtained about parking availability, both outside and inside, and regarding traffic and road conditions.

SMART PARKING TECHNOLOGY

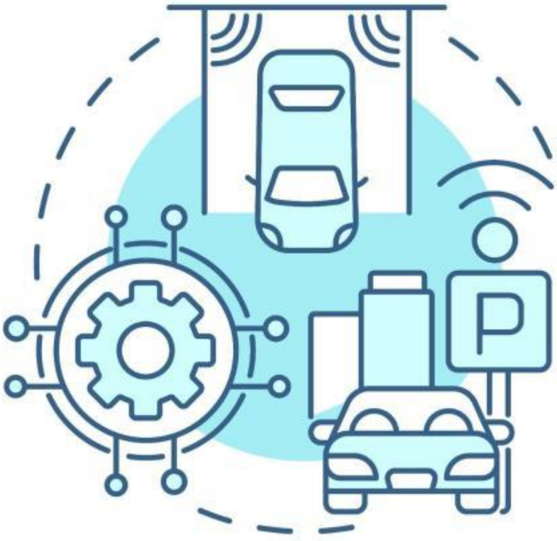
Various devices and processes form the structure of smart parking, acting as parking space detectors. On the one hand, the **deployment of sensors and/or cameras**, which record and process data and images to provide real-time traffic occupancy data for the area we are heading to.

An IoT cloud-based system, on the other hand, allows these devices to be connected and the data to be centralized. The data are then analysed using big data in order to calculate the availability of on-street parking spaces or spaces in public and private parking facilities.

Smart parking maps

If we want even more accurate information about how likely we are to find an on-street parking space, we don't always have to use an app. Functionalities already available on our devices such as Google Maps provide us with real-time traffic data and the **likelihood of parking in these areas.** This service and other maps update the information the closer we get to our selected destination.

Smart signage

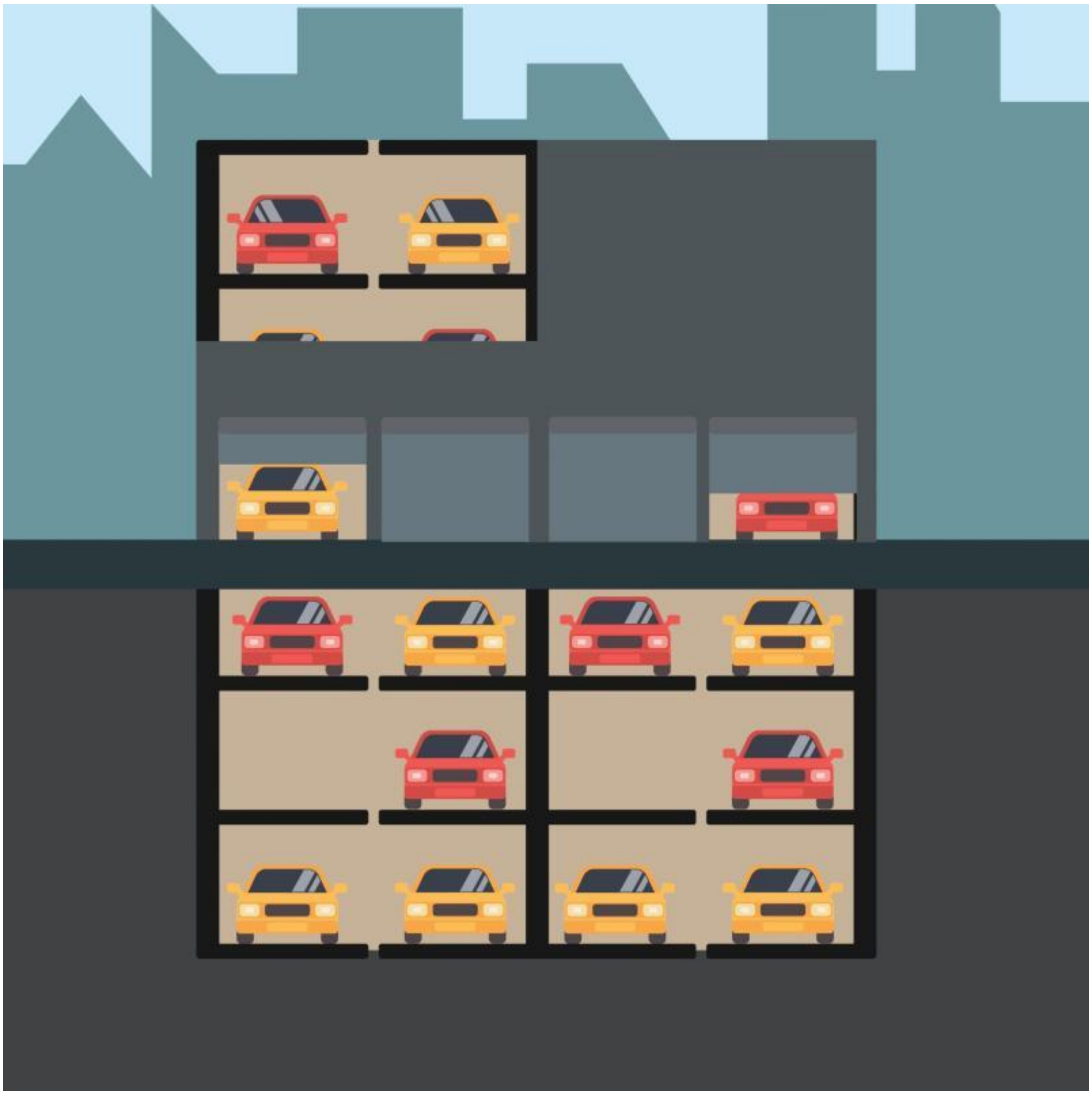


Smart technologies are also being used in road-sign systems with the aim of **increasing safety and helping to coordinate pedestrian and vehicle traffic more efficiently.** Examples include traffic lights and pedestrian crossings that change color or light up depending on real-time or estimated traffic volumes, such as peak hours.

Smart detectors for vehicles

Knowing exactly how many vehicles are located in a parking lot at any given time is the basis of smart parking. This car parking monitoring system is made up of sensor systems including **dual channel loop detectors, ultrasonic vehicle presence sensors or LiDAR vehicle sensors.** They detect whether the parking space is free/occupied, they identify if a parking garage is full and provide an accurate location of vehicles, respectively.

Sensors to detect parking spot occupancy



In this case, these sensors **detect available parking spaces,** facilitating the task for drivers looking for vacant parking spots in closed spaces. Thanks to the incorporation of LED indicators, drivers can see how many parking spaces are available, with red or green light signs indicating whether the parking space is currently used or is free for parking.

IoT traffic light. Sensors installed in strategic locations can use IoT technology to collect data on congestion, moving vehicles away from these locations. IoT Big Data_solutions can analyze this information, determine alternative routes, and improve traffic signaling to reduce congestion. Roadside lights can also work by the weather sensors installed on them. With a light control system, roadside lighting will change with the onset of day or night and when weather conditions require it.

Smart Traffic Management: Optimizing Your City's Infrastructure Spend

Smart Traffic Management Systems are technology solutions that municipalities can integrate into their traffic cabinets and intersections today for fast, cost-effective improvements in safety and traffic flow on their city streets. What's more, deploying these systems today, or upgrading your city's existing Intelligent Transportation Systems (ITS) infrastructure can create huge efficiencies and cost savings, while massively improving system reliability, all of which have excellent ROI.

These systems utilize sensors, cameras, cellular routers, automation to monitor, and automatically direct traffic and reduce congestion. The right technology solution can be scaled to any size and painlessly upgraded at any time. Simultaneously, these technology solutions prepare Smart Cities for coming technology evolutions, including Connected Vehicle and the full deployment of 5G networks.

Why Evaluate Smart Traffic Technology?

Budgets for public infrastructure are always tight, and constructing roads and bridges is always expensive. Smart Traffic Management Systems help municipal and regional transportation departments to cope with the situation — quickly and cost-effectively. Integrating smart traffic technology helps them affordably get better performance from their existing infrastructure.

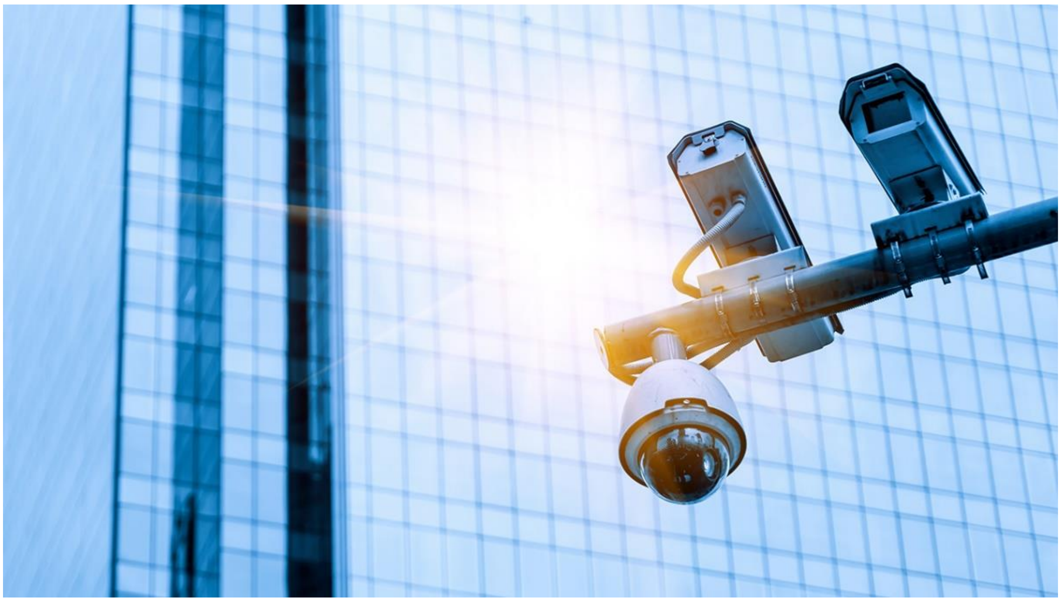
Let's explore how augmenting and retrofitting infrastructure can dramatically improve the efficiency and safety of existing traffic networks.

Smart Traffic Management: Smart Cities Do More With Less

The problems plaguing our streets and highways are well known. Traffic slow-downs can cause debilitating congestion and add to urban air pollution. Businesses suffer from delivery delays and lost productivity. Emergency vehicles are slowed down by bottlenecks, potentially putting lives at risk. And all of it diminishes the city's overall quality of life.

Meanwhile, cities and regional governments continually ask their traffic management teams, civil engineers and highway maintenance crews to do more with less. In the face of these challenges, innovative cities — "Smart Cities" — are using a coordinated array of hardware, software and cloud solutions to increase traffic flow and improve safety. Smart Traffic Management Systems, which are included in the umbrella of “intelligent transportation systems” and sometimes called “intelligent traffic management,” are automated systems that incorporate the latest advances in Internet of Things (IoT) technology.

These systems can optimize traffic flow and enhance safety by using sensors, cameras, routers and cellular technology to dynamically adjust control mechanisms such as traffic lights, freeway on-ramp meters, bus rapid transit lanes, highway message boards and even speed limits.



Today, Smart Traffic Management Systems make it possible to increase the capacity of city streets without actually adding new roads. With the advent of connected vehicle technology, these systems will also be able to directly control vehicles when needed — braking them in intersections, for example, to prevent accidents with pedestrians or other vehicles. Smart Cities are deploying these systems now to be prepared when the vehicle technology is fully tested and deployed.

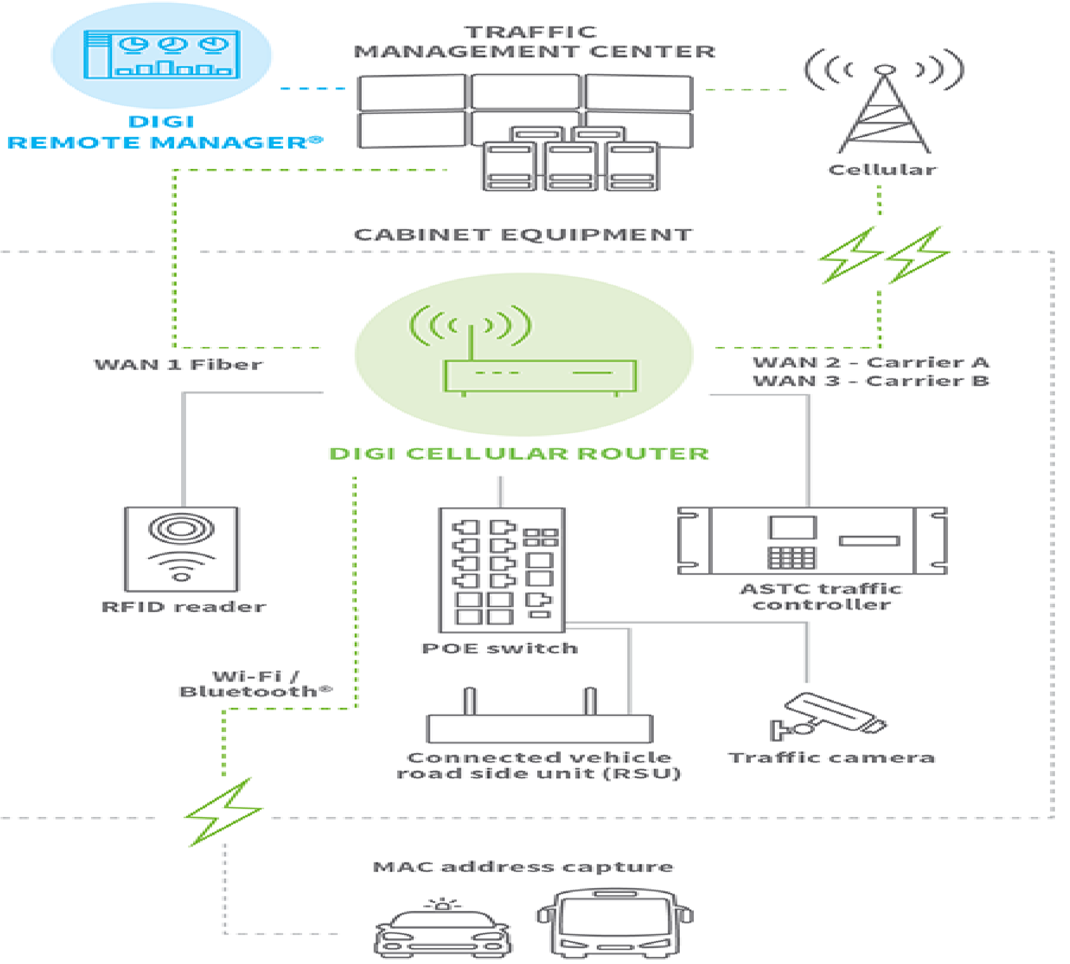
The Enabling Technology Behind Smart Traffic Management

All of this connectivity requires sophisticated hardware and software. Digi offers secure, scalable, high-speed connectivity to address the range of traffic management requirements. Designed to operate optimally regardless of moisture, vibration and temperature extremes, Digi industrial and transit routers are installed in Smart Cities around the world.

Some of the key functionality cities achieve with these systems include the following:

- Congestion detection: With cameras and sensors constantly monitoring intersections, technicians can monitor the entire city from the city's traffic management center.

- Adaptive control: Congestion detection also enables adaptive control, which causes dynamic adjustments to systems including traffic lights, on-ramp signaling, and bus rapid transit lanes.
- Connected vehicle: This up-and-coming technology enables vehicles to communicate directly with intersections. The Smart Traffic Management can include a connected vehicle roadside unit for this purpose.
- Emergency routing: A critical application of the Smart Traffic Management System is the ability to give priority access to police, fire and ambulance services.



All of these functions require monitoring, detection, reliable high-speed data transfer and automation. One purpose-built Digi cellular router can provide the communications backbone to manage all of these systems.

Examples of Digi's communications solutions for traffic management include the following:

- **FirstNet Ready™ Digi TX54 Router for Critical Connectivity**
  Digi TX54 is a secure, high-performance cellular router that has been approved for use on the FirstNet® communications platform, the high-quality spectrum developed for police, fire, ambulances and other emergency services. Digi WR54 serves as the primary router and workhorse managing communications and providing cellular redundancy to ensure uptime. The Digi WR54 is secure by design, with Digi TrustFence®, a sophisticated suite of integrated hardware and firmware features. Some of the key features include link failover and load balancing across dual 600 Mbps CAT 12 cellular connections, and Ethernet, serial, Wi-Fi and Bluetooth for connecting a range of assets.
- **Digi WR31**
  Digi WR31 is a cellular router that collects data from roadside sensors, traffic controllers and cameras positioned at traffic intersections, highway ramps and other points, and can be deployed in traffic management systems that do not require the full functionality of the Digi WR54. Built to endure the harsh conditions of outdoor cabinet enclosures, Digi WR31 sends data via high-speed 4G LTE links to the traffic management center, where it can be analyzed automatically to optimize traffic flow. Digi TransPort WR31 offers software-defined, multi-carrier networking and drop-in connectivity to help reduce downtime and bring distributed sites online faster. It even sends an alert if a traffic control cabinet is left open or unlocked. The Digi TransPort WR31 is also suitable use with on-site solar panels—reducing cost and increasing deployment flexibility.
- **Digi IX14**
  This rugged, industrial LTE router connects meters and remote sensing equipment to traffic management centers, and is a low-cost system for the most cost-conscious traffic management operations. Digi IX14 is also secured by Digi TrustFence, which protects the router and ensures the reliability and integrity of its security functions. Digi IX14 also includes built-in mechanisms that enable it to be updated and managed remotely.
- **Digi Remote Manager — for configuration, remote access and control**
  The Digi cellular routers in your traffic management network can be quickly configured using Digi Remote Manager®, and then managed remotely from a single platform. This secure, hosted application lets administrators monitor and troubleshoot their systems as well as perform mass configurations or simultaneous software updates. Digi Remote Manager offers a single point of command and control, along with system-wide monitoring that includes dashboards to

view performance metrics, and automated alerts for specified conditions and thresholds. Digi Remote Manager can generate reports ranging from daily summaries to detailed period-based metrics. It also helps with the integration of legacy systems into the traffic management network. For example, new devices added to the system can be configured quickly in Digi Remote Manager from saved configurations to either swap out devices or scale up the installation with more devices.

Digi technology solutions integrate with existing traffic management systems to provide critical communications, security, reliability, and device management. Digi also offers professional services to integrate and install the systems and develop applications for specific requirements. With thousands of installations across the U.S., our teams have deep experience in tailoring and installing these systems on tight timelines.

Improving Commuter Traffic and Emergency Preparedness

Daily traffic congestion may be one of your city's biggest headaches. Ensuring emergency vehicles can rapidly and effectively reach their destinations is a critical metric of success for any traffic management system as well. In addition to handling these urban challenges, city managers also know that disaster preparedness is critical in the event of city-wide events.

During a large scale public emergency, such as a hurricane, tornado or other major disaster, civilian cell phone use can jam cellular networks, jeopardizing communications with first responders and city leaders.

FirstNet is a U.S. nationwide wireless network designed to provide voice, text and data communications for emergency responders — the *primary* category of public safety entities —as well as "extended primary" services, which include critical infrastructure such as power grids, commuter rail, and water and sewage systems. Digi is building cellular systems to support smart cities in their critical disaster preparedness planning, such as the FirstNet Ready™ Digi WR54 cellular router. This designation means that it has been tested and certified to meet the strict criteria for devices designed for these critical situations.

Preparing for 5G, Connected Vehicles and More

Techology is always evolving and the future is on the horizon, even as organizations and municipalities make their best choices for infrastructure maintenance and enhancements today. As a technology leader in the networking and IoT space, Digi develops mission critical solutions that are designed for longevity, scalability and evolution to support enterprises and smart cities in long-term planning. For information on the integration of LTE and 5G technologies and networks, and to start your 5G planning, be sure to visit our 5G information page.

Connected vehicle technology is one example of an innovation that will have a huge impact on smart city technology planning. Under trials at test sites, connected vehicle technology will enable vehicle-to-intersection communications and automated braking. As networks become faster, and latency lowers, the ability to make instantaneous, automated decisions for driver and pedestrian safety will increase. Many Smart Cities are integrating the infrastructure techology today to be prepared for connected vehicle at its earliest availability. In the foreseeable future, technologies such as machine learning will also be deployed as it is tested for efficacy in identifying objects and making rapid decisions to improve city safety.

Reducing Congestion without Widening Roads

In spite of constrained budgets for infrastructure upgrades, Smart Cities around the world are using smart traffic management technologies to improve the overall performance of their traffic networks, with high performance, secure, redundant and reliable communications solutions that can help streets and highways carry more traffic in greater safety.

At Digi, we have teams with deep experience in transit and traffic management, as well as fleet technology, telematics and system integration to support municipalities in making their smartest infrastructure and city management choices.

Home Automation

Smart home automation devices connect appliances, switches, and gadgets to a central hub, enabling you to control those devices in secure and convenient ways. Command groups of lights to turn on or off when you open the front door, receive alerts when a leak springs, or program your thermostat to work with your fans and AC when the room gets hot. Smart home automation devices make your home more comfortable, more convenient, and more secure than ever.

Home Controls offers an enormous variety of home automation products, from complete smart home systems to smart switches, from top-of-the-line brands (Leviton, Elk, Qolsys) to DIY easy-to-use options (X10, Skylink, Z-Wave) and multiple products that are compatible with Alexa, Google Assistant, and Apple Homekit. We carry everything you need to control, monitor, and secure your home, including home automation hubs and controllers, remote control modules and smart switches, all types of sensors, and much more.

Intrusion detection system

An intrusion detection system (IDS; also intrusion prevention system or IPS) is a device or software application that monitors a network or systems for malicious activity or policy violations.Any intrusion activity or violation is typically reported either to an administrator or collected centrally using a security information and event management (SIEM) system. A SIEM system combines outputs from multiple sources and uses alarm filtering techniques to distinguish malicious activity from false alarms.IDS types range in scope from single computers to large networks. The most common classifications are network intrusion detection systems (NIDS) and host-based intrusion detection systems (HIDS). A system that monitors important operating system files is an example of an HIDS, while a system that analyzes incoming network traffic is an example of an NIDS. It is also possible to classify IDS by detection approach. The most well-known variants are signature-based detection (recognizing bad patterns, such as malware) and anomaly-based detection (detecting deviations from a model of "good" traffic, which often relies on machine learning). Another common variant is reputation-based detection (recognizing the potential threat according to the reputation scores).

Introduction

Incredible developments in the routine use of electronic services and applications have led to massive advances in telecommunications networks and the emergence of the concept of the Internet of Things (IoT). The IoT is an emerging communications paradigm in which devices serve as objects or “things” that have the ability to sense their environment, connect with each other, and exchange data over the Internet [1, 2]. By 2022, one trillion IP addresses or objects will be connected to the Internet through IoT networks [3].

The IoT paradigm has recently been used in creating smart environments, such as smart cities and smart homes, with various application domains and related services. The goal of developing such smart environments is to make human life more productive and comfortable by solving challenges related to the living environment, energy consumption, and industrial needs [4]. This goal is directly reflected in the substantial growth in the available IoT-based services and applications across different networks. For example, the Padova Smart City in Italy is a successful example of a smart city based on an IoT system [5].

Smart environments consist of sensors that work together to execute operations. Wireless sensors, wireless communication techniques, and IPv6 assist in the expansion of smart environments. Such environments are wide ranging, from smart cities and smart homes to smart healthcare and smart services. The integration of IoT systems and smart environments makes smart objects more effective. However, IoT systems are susceptible to various security attacks, such as denial-of-service (DoS) attacks and distributed denial-of-service (DDoS) attacks. Such attacks can cause considerable damage to the IoT services and smart environment applications in an IoT network. Consequently, securing IoT systems has become a major concern [1]. For example, on Friday, October 21, 2016, a series of DDoS attacks were launched across the US that exploited the security vulnerabilities in IoT systems [6]. These attacks affected IoT devices, websites and online services such as Twitter, Netflix, and PayPal.

An intrusion detection system (IDS) is a security mechanism that works mainly in the network layer of an IoT system. An IDS deployed for an IoT system should be able to analyze packets of data and generate responses in real time, analyze data packets in different layers of the IoT network with different protocol stacks, and adapt to different technologies in the IoT environment [3]. An IDS that is designed for IoT-based smart environments should operate under stringent conditions of low processing capability, fast response, and high-volume data processing. Therefore, conventional IDSs may not be fully suitable for IoT environments. IoT security is a continuous and serious issue; thus, an up-to-date understanding of the security vulnerabilities of IoT systems and the development of corresponding mitigation approaches are required.

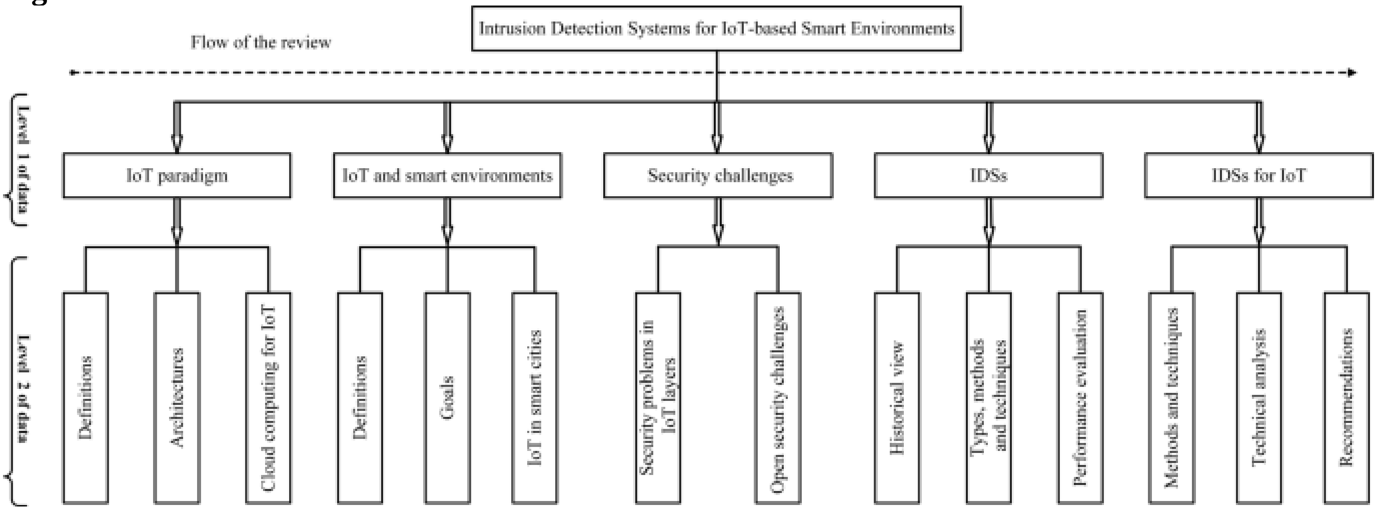
This article offers a comprehensive review of IDSs as a security solution for IoT-based smart environments. The primary goal of this study is to present the most recent designs and approaches for IDSs operating in IoT-based environments. Although related surveys have been published in the literature [3, 7], this article focuses on the important factors that affect IDS performance in smart environments, such as the detection accuracy, false positive rate, energy consumption, processing time, and performance overhead. In addition, this article introduces a solid foundation for the development of IDSs for IoT-based smart environments.

This study offers multiple key contributions. First, a full preliminary analysis of IoT systems, smart environments, and IDSs is presented. Second, the study confirms that traditional IDSs cannot satisfy IoT security requirements due to the large diversity of IoT networks and protocols. For instance, IPv6 over low-power wireless personal area networks (6LoWPAN) is not a protocol that is used in traditional telecommunications networks. Third, the common features that can be ported from traditional IDSs to IoT-based IDSs are emphasized. This third contribution emerges from the integration of the previous surveys [3, 7] to summarize the features, advantages and disadvantages of all IDSs designed for IoT-based systems. Fourth, this work introduces a future outlook on IDSs for IoT environments with a focus on the strengths and weaknesses of the current IDSs. Additionally, this study presents new recommendations for designing IDSs that satisfy the security requirements of IoT-based smart environments.

This survey focuses on IDSs for the IoT paradigm, independent of any specific technology or protocol; however, readers who are interested in learning more about IoT enabling technologies and protocols such as low-power wide-area network (LPWAN) technologies, long range (LoRa) technology, the low power WAN protocol for Internet of Things (LoRaWAN), the 6LoWPAN protocol, or the constrained application protocol (CoAP) may refer to [8–11] for further details.

The remainder of this paper is organized as follows. "The IoT paradigm" section discusses various definitions and architectures relevant to the IoT context. This section also highlights the importance of cloud computing systems for IoT-based smart environments and the challenges of applying this combination of systems in the real world. Definitions, goals and challenges related to smart environments, with a focus on smart cities, are discussed in "IoT and smart environments" section. "Security challenges in IoT-based smart environments" section reviews the security challenges in IoT-based smart environments in relation to the various layers of the IoT architecture and highlights some practical open challenges facing real-world IoT networks. "Intrusion detection systems (IDSs)" section provides preliminary information about the definitions relevant to IDSs, the different types of IDSs and the detection techniques used in these systems. A survey of IoT-oriented IDSs that either can be applied in or are specifically designed for smart environments is presented in "IDSs designed for IoT systems" section."Discussion and future outlook" section discusses recommendations concerning IDSs implemented for IoT-based smart environments. Finally, conclusions and plans for future work are reported in "Concluding remarks" section. The organization of this paper is presented visually in Fig. 1.

Fig. 1



The flow of this article, separated into two levels; level 1 describes the main topics, and level 2 describes the detailed points

Full size image

The IoT paradigm

The IoT concept has been established since the founding of the Auto-ID Center at the Massachusetts Institute of Technology (MIT) in 1999. The Auto-ID Center created the electronic product code (EPC) number, which depends on radio frequency identification (RFID), in 2003. This idea is the crucial technology of the IoT [12].

However, the IoT is a well-established paradigm, and it is defined in several ways from various perspectives. Thiesse et al. [13] defined the IoT as consisting of hardware items and digital information flows based on RFID tags. The IoT definitions and architectures provided by various standards and industrial organizations will be described in the following.

The Institute of Electrical and Electronics Engineers (IEEE) defines the IoT as a collection of items with sensors that form a network connected to the Internet [12, 14]. The International Telecommunication Union (ITU) defines the IoT through three dimensions, as a network that is available anywhere, anytime, and by anything and anyone [15]. The European Telecommunications Standards Institute (ETSI), rather than using the expression “Internet of Things (IoT)”, defines machine-to-machine (M2M) communications as an automated communications system that makes decisions and processes data operations without direct human intervention [16].

The Coordination and Support Action for Global RFID-related Activities and Standardisation (CASAGRAS) project has created a new concept of the IoT that encompasses two viewpoints: the connection of physical objects with virtual objects over a global network without any human intervention to the greatest extent possible [17] and the incredible increase in IoT applications within traditional networks due to the extent of IoT marketing [17].

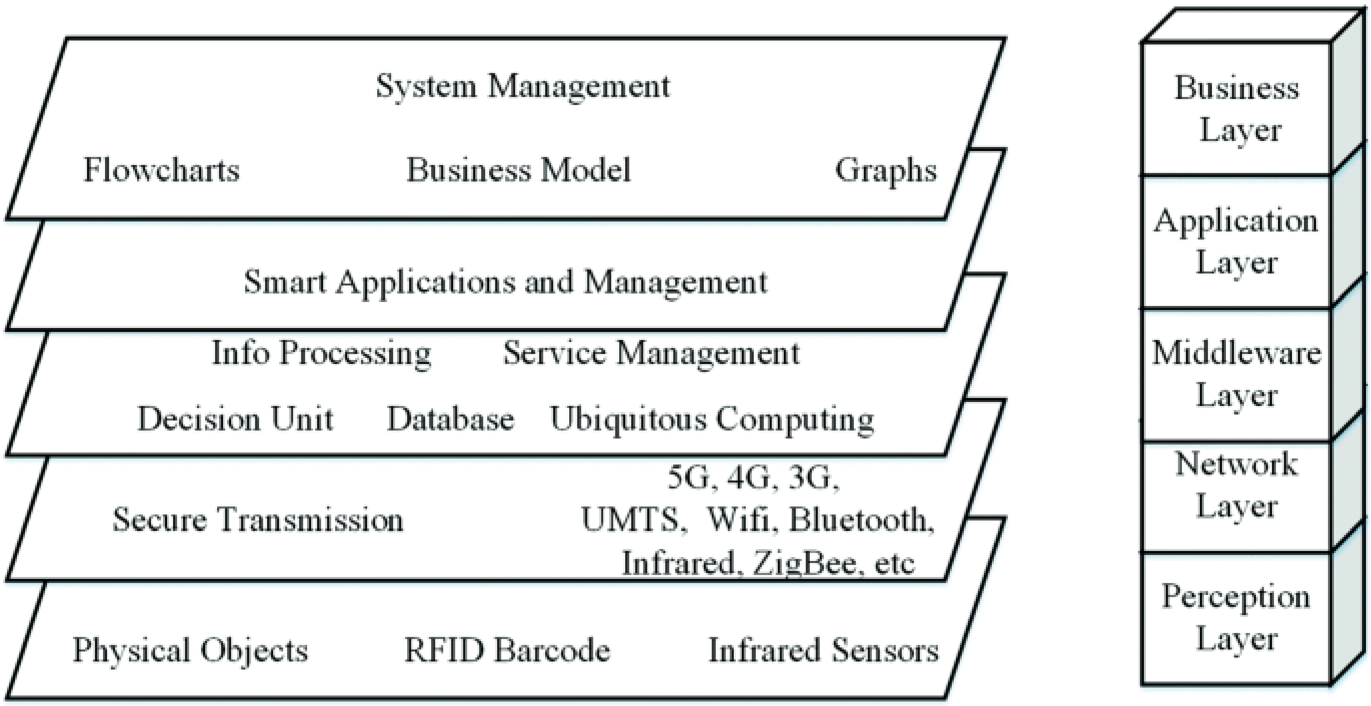
Moreover, Cisco, an industrial organization, works on IoT technology under the title of the Internet of Everything (IoE). Cisco has summarized the IoE concept as a network that consists of people, data, things, and processes. Thus, information and actions are created in and moved through this network [18].

IoT system architectures

Regarding IoT design, IEEE is working on a project (IEEE P2413) to determine the IoT architectural framework. The scope of this project is to describe the IoT domains and the various applications in these domains [19]. This IoT architecture is divided into three layers: the application layer, the networking and data communications layer, and the sensing layer.

According to [20–22], the general architecture of the IoT is divided into five layers that span three domains, namely, the application domain, the network domain, and the physical domain; thus, the IoT can be customized to fit the needs of different smart environments. The application domain encompasses management and utilization. The network domain is responsible for data transmission. The physical domain is responsible for information collection. The layers of the general IoT architecture are shown in Fig. 2. The functionality of the different layers is discussed in the following.

Fig. 2



The general architecture for the IoT. The general architecture for the IoT, which consists of five layers according to [22]

Full size image

The perception layer is a hardware layer that consists of sensors and physical objects in different forms. These hardware elements provide identification, information storage, information collection, and information processing. The information output from this layer is sent to the next layer (the network layer) to be transmitted to the processing system [22].

The network layer is a transmission layer that transfers the information from physical objects or sensors to the processing system over secure lines using a communication system. This communication system can be either wired or wireless and can be based on different technologies, depending on the physical object or sensor components. The information output from this layer is sent to the next layer (the middleware layer) [22].

The middleware layer is responsible for service management over IoT devices to create connections between IoT devices that provide the same service. Moreover, the middleware layer stores the information coming from the network layer in a database to facilitate decision-making on the basis of information processing operations [22].

The application layer is responsible for the global management of IoT applications. The application layer depends on the information processed in the middleware layer. Moreover, the application layer depends on the specifics of the different implemented IoT applications, such as smart industry, building, city, and health applications [22].

The business layer is also responsible for the global management of IoT applications as well as service management over IoT devices. The business layer creates a business model that depends on the information processed in the application layer and on the analysis of the results of these information processing operations [22].

Cloud computing and the IoT

IoT systems connect an enormous number of devices and sensors exchanging an enormous amount of data and supporting a massive number of services. The management and analysis of these data pose certain special requirements, such as powerful processing, massive storage and high-speed networking capabilities [23].

Cloud computing offers high computational power, a massive storage capacity, and configurable resources with virtualization capabilities for manipulating the large amounts of data collected from IoT-based smart environments. With the integration of cloud computing systems and IoT-based smart environments, smart things can be easily accessed and managed at any time and place, and better services can be provided through the IoT model [23, 24].

According to [25], one of the important challenges in employing a cloud computing system for the IoT is the synchronization between different cloud vendors. A second challenge is achieving compatibility between general cloud service environments and IoT requirements. Security challenges are the main factor hindering the adoption of cloud computing by businesses and government organizations [26]. Thus, the ability to respect the necessary security constraints to fulfill the needs of the IoT in a cloud computing platform is a vital requirement. A robust and efficient security solution such as an IDS is one possible option. Moreover, standardization, enhancement, and management for the deployment of IoT systems and their connection to the cloud are additional challenges that should be taken into consideration.

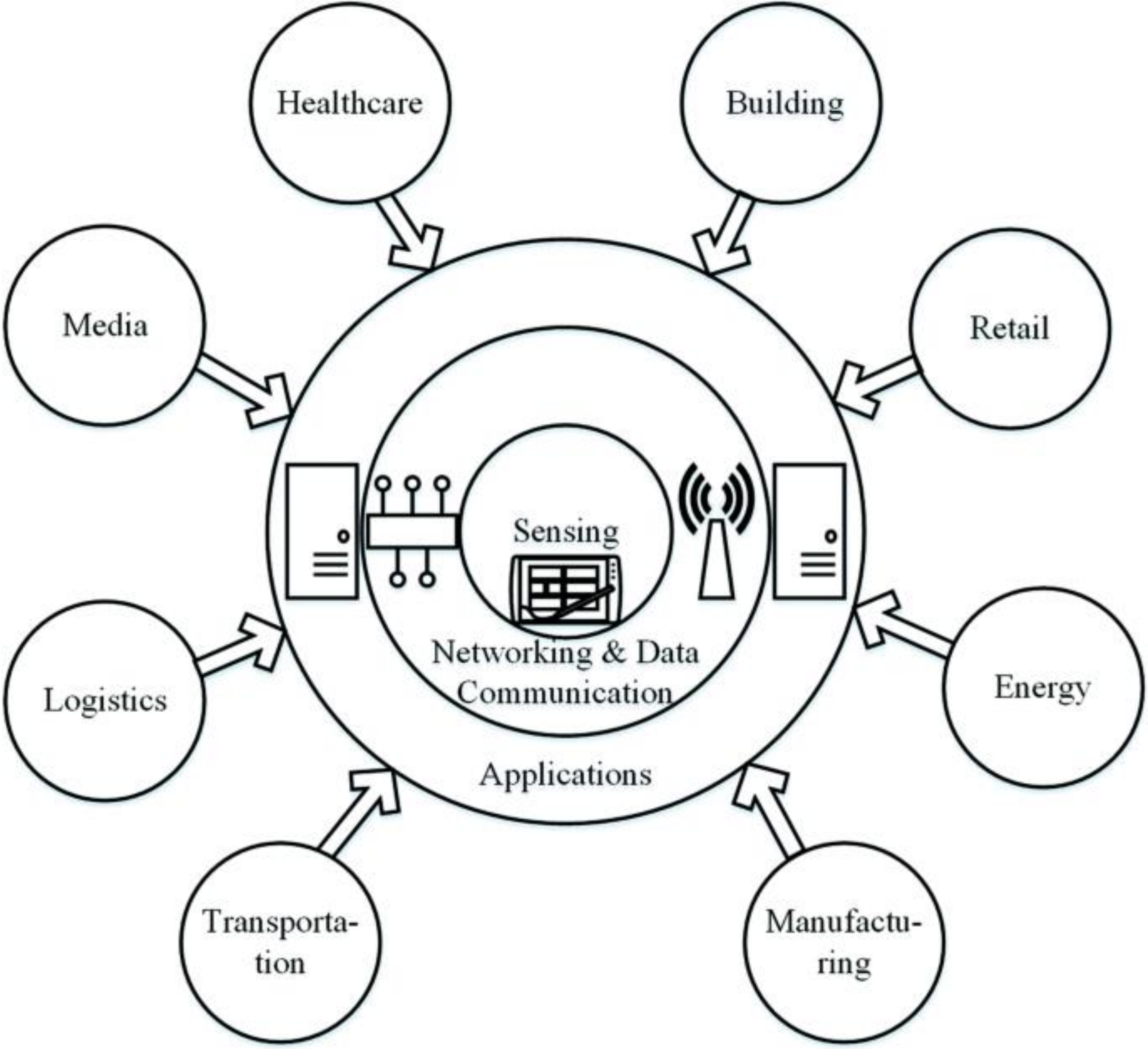
IoT and smart environments

The objective of smart environments is to make human life more comfortable and more efficient by using sensors. IoT-based smart environments enable the effective realization of smart objects. By means of an IoT network, sensors can be monitored and controlled remotely. According to Navigant Research, the global smart city services market is expected to increase from 93.5 billion US dollars in 2017 to 225.5 billion US dollars by 2026 [27].

Ahmed et al. [28] state that “The term smart refers to the ability to autonomously obtain and apply knowledge, and the term environment refers to the surroundings”. A smart city is one type of smart environment. The core element of a smart city is an integrated information center operated by the IoT service provider, which provides information on services such as electricity, water, and gas.

Smart health, smart industry, smart buildings and smart homes are other types of smart environments. The objective of such smart environments is to provide services via smart methods based on the information collected by IoT-enabled sensors. The architecture of such IoT-based smart environments is shown in Fig. 3.

Fig. 3



IoT-based smart environments. The architecture of the IoT and the extent of the IoT market according to [14]

Full size image

Smart environments based on the IoT paradigm have certain special characteristics, and hence, special needs arise in the deployment of such environments. For instance, remote monitoring and remote control capabilities are required to allow smart objects to collect and process data and to execute operations remotely. Moreover, the ability to make decisions is an important characteristic in such a system. A smart object should be able to make intelligent decisions without human intervention by using data mining and other techniques for extracting useful data.

By virtue of these characteristics, smart environments offer certain features that can be used to enhance the quality of service (QoS) of user applications. Real-time information is one of these features. Smart objects can collect and analyze data and make intelligent decisions in real time. Moreover, the cost-effectiveness of cloud applications can be used to increase the QoS of smart environment applications. The integration of smart and IoT environments offers new opportunities with respect to the QoS of services and applications.

IoT technology for developing smart cities

Many national governments are working on the information and communication technology (ICT) infrastructure to solve the problems arising in traditional public management affairs. One of the most modern and effective solutions is to establish a smart city [5]. The smart city concept is one facet of the idea of smart environments.

There are many benefits of converting traditional public services and resources into a form that takes advantage of the smart city concept, including increasing the quality of public services and reducing the operating costs of public administration [29]. However, the management and execution of public services in a smart city require a powerful network, such as an IoT network.

Additionally, there are many barriers to the establishment of an IoT-based smart city. The novelty, complexity and technical challenges of IoT systems present the greatest difficulty. Furthermore, in the absence of widely accepted definitions for smart city operations, political and financial barriers prevent the smart city concept from being effectively applied.

The Padova Smart City in Italy is a successful example of a smart city that has overcome these barriers. The main goal of establishing the Padova Smart City is to develop ICT solutions for public administration systems using different types of data and technology [5].

The implementation of the IoT paradigm for creating smart environments, particularly smart cities, faces several technical challenges. Among these, precision, latency and available bandwidth have important effects in many smart environments, such as industrial and healthcare environments. Because of the need to support an increasing number of users and smart objects in IoT networks and the corresponding generation of increasingly large amounts of data, scalable computing platforms, such as cloud computing, are necessary. Such platforms can improve the performance of data management services in IoT systems and the QoS of smart environment applications [30].

Security challenges in IoT-based smart environments

The security of IoT systems is a serious issue due to the increasing numbers of services and users in IoT networks. The integration of IoT systems and smart environments makes smart objects more effective. However, the impacts of IoT security vulnerabilities are very dangerous in critical smart environments used in fields such as medicine and industry. In IoT-based smart environments without robust security systems, applications and services will be at risk. Confidentiality, integrity, and availability are three important security concepts of applications and services in IoT-based smart environments; thus, to address these concerns, information security in IoT systems requires greater research focus [2]. For example, IoT-based smart homes face security and privacy challenges that span all layers of the IoT architecture [31].

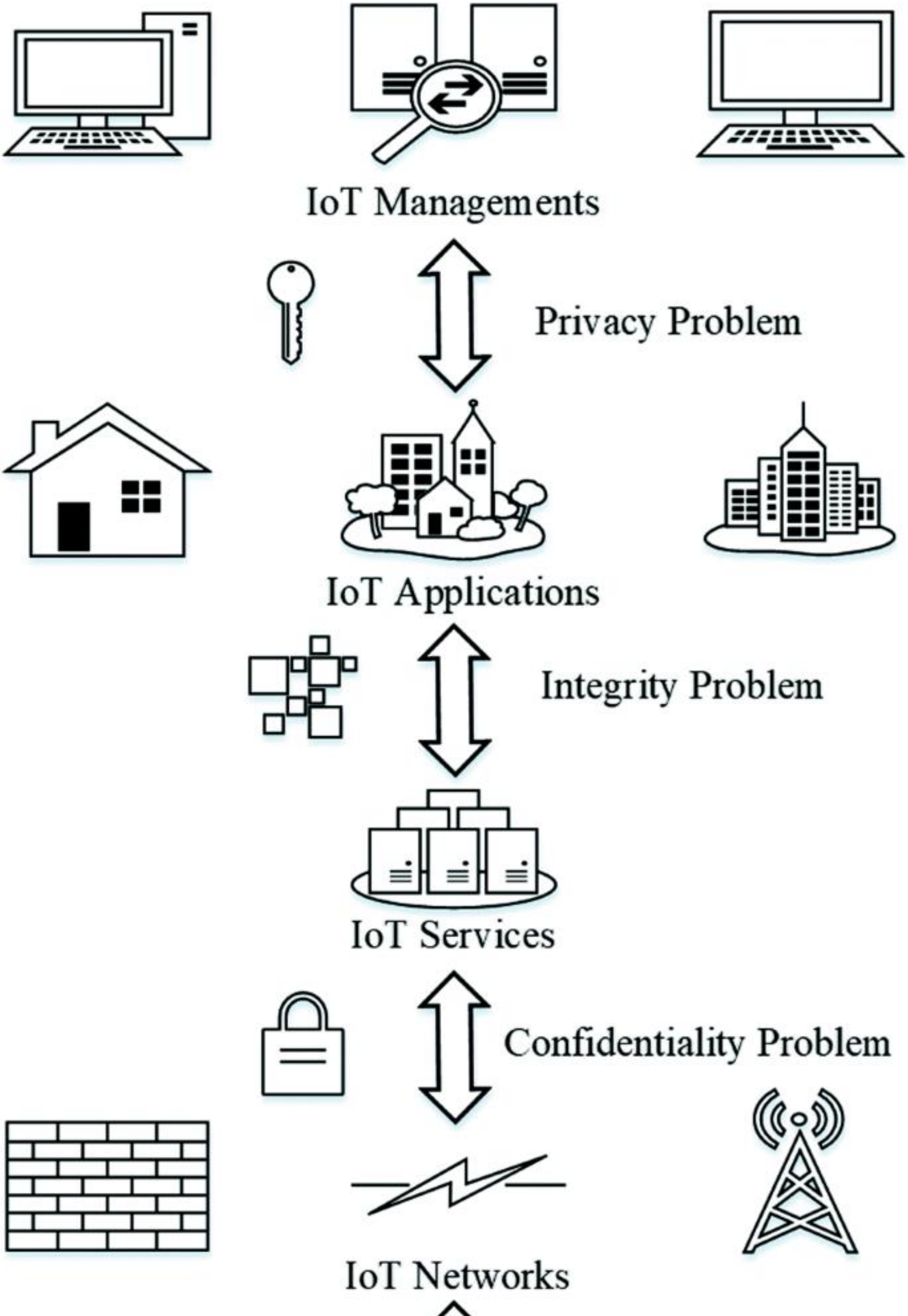
The creation of smart environments in the real world faces two notable barriers: the security of IoT systems and the complexity and compatibility of IoT environments. Attacks such as DoS or DDoS attacks on IoT networks affect IoT services and thus affect the services provided by smart environments.

Researchers study the security challenges of the IoT from many different points of view, one of which is the security vulnerability of IoT communication protocols [32]. This survey focuses on IDSs for the IoT paradigm, independent of any specific protocol; thus, this study focuses on the security challenges facing IoT systems on the basis of the IEEE definition and the general IoT architecture.

The security challenges in IoT systems are related to security issues arising in the different IoT layers. Physical damage, hardware failure, and power limitations are challenges faced in the physical layer. DoS attacks, sniffing, gateway attacks, and unauthorized access are challenges relevant to the network layer. Malicious code attacks, application vulnerabilities, and software bugs are challenges faced in the application layer [33].

According to [34], the security-related problems of any IoT system can be categorized into four types: authentication and physical threats, confidentiality risks, data integrity issues and privacy problems. The relations between these groups are shown in Fig. 4. The security problems arising in the different IoT layers are concisely discussed below.

Fig. 4


IoT Managements
Privacy Problem
IoT Applications
Integrity Problem
IoT Services
Confidentiality Problem
IoT Networks

The security challenges in the different IoT layers. The four types of security problems arising in the IoT model are associated with the different layers of the IoT architecture

Full size image

- The authentication-related problem and physical threats are the first challenges that affect an IoT system. The perception layer includes many IoT devices, such as sensors, that depend on their own security systems; thus, they are susceptible to physical attacks.
- Confidentiality-related risks arise between IoT devices and the gateways in the network layer. The resource-constrained nature of the low-level devices in IoT systems poses an indirect challenge with regard to the confidentiality of data transmission in IoT networks [35].
- The third class of security challenges concerns the data integrity between services and applications. Data integrity problems emerge when spoofing attacks or noise affect an IoT system. DoS, DDoS, and probe attacks are arbitrary attacks that can harm IoT applications and services.
- The challenges of the fourth type are related to privacy. Information privacy is an important aspect of security in IoT systems [36]. Different IoT components use different types of object identification technologies; thus, every object has its own identification tag, which carries personal, location and movement information. Managing and monitoring the applications and services in an IoT system mean placing information privacy at risk; for example, using a system based on a deep packet inspection technique for trusted operations within an IoT system is considered to be a violation of information privacy [37]. Any intrusive accesses to the management system without permission threaten the information privacy of the IoT users [34].

Real-world applications of IoT systems face many open challenges. The open security challenges affecting IDS operations identified by [20, 22, 33, 38] are discussed below.

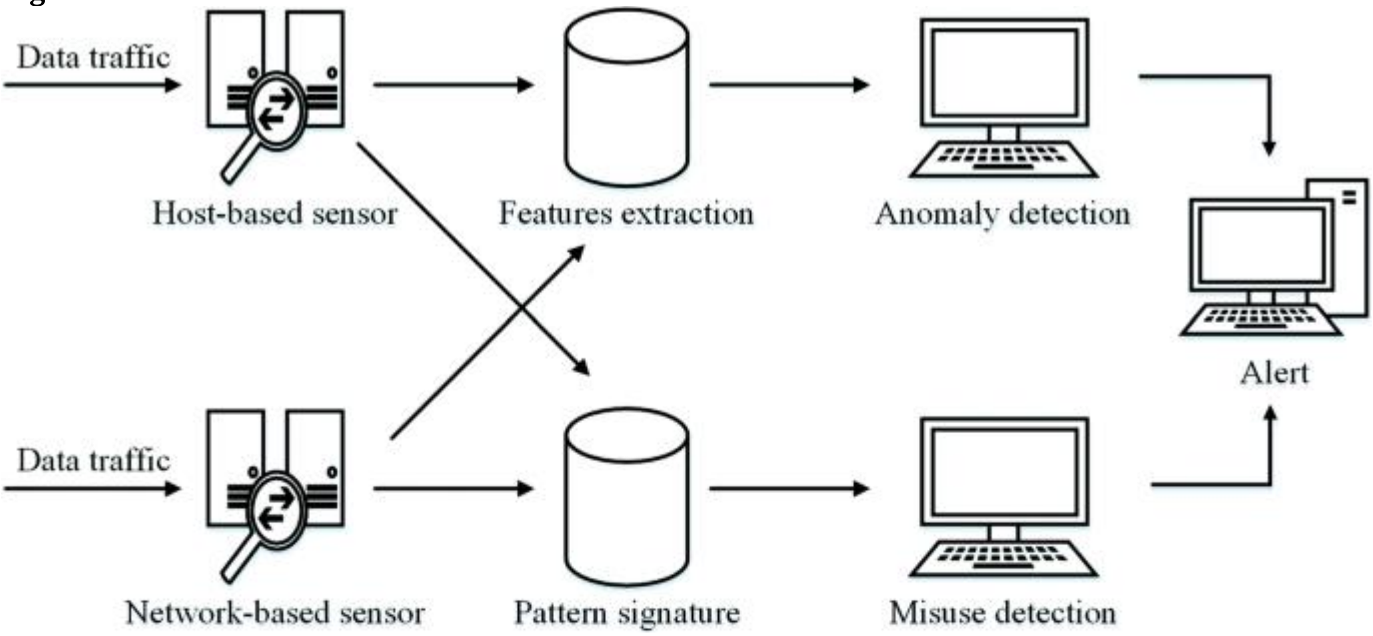
- A smart environment that integrates IoT technology is considered to be a complex system because it consists of different products from different companies based on different technologies that do not share a universal language. Therefore, standardization is another important aspect of security in IoT systems. Creating a standard IoT architecture based on one standard technology for all vendors and manufacturers would enhance the interoperability of the security functionalities of all objects and sensors in an IoT system. The success of this integration will depend on collaboration among companies to create a universal standard. Such standardization will greatly facilitate IoT network security.
- A single successful penetration of one or more end devices can threaten the security of an entire IoT system and cause harm to its applications and services, especially from an industrial point of view [39]. Thus, the implementation of a strong security mechanism in an IoT system depends on the strength of the security for individual IoT devices, which in turn depends on power and memory factors. Consequently, power and memory limitations are considered to pose indirect security challenges in IoT systems. To address these challenges, lightweight security solutions and lightweight encryption and decryption methods are required. These solutions and methods must be applicable in different IoT domains and must satisfy the security requirements without affecting the QoS.

Intrusion detection systems (IDSs)

IDSs: a historical overview

Monitoring and analyzing user information, networks, and services through passive traffic collection and analysis are useful tools for managing networks and discovering security vulnerabilities in a timely manner [40, 41]. An IDS is a tool for monitoring traffic data to identify and protect against intrusions that threaten the confidentiality, integrity, and availability of an information system [42]. The operations of an IDS are schematically illustrated in Fig. 5.

Fig. 5



IDS operations. IDS operations can be divided into the monitoring stage, the analysis stage and the detection stage

Full size image

The operations of an IDS can be divided into three stages. The first stage is the monitoring stage, which relies on network-based or host-based sensors. The second stage is the analysis stage, which relies on feature extraction methods or pattern identification methods. The final stage is the detection stage, which relies on anomaly or misuse intrusion detection. An IDS captures a copy of the data traffic in an information system and then analyzes this copy to detect potentially harmful activities [43].

The concept of an IDS as an information security system has evolved considerably over the past 30 years. During these years, researchers have proposed various methods and techniques for protecting different types of systems using IDSs. In 1987, Denning presented an intrusion detection model that could compare malicious attack behavior against the normal model for the system of interest [44].

In 2000, Axelsson [45] surveyed 20 research projects on IDSs. He listed fourteen IDSs relying on host-based methods, two IDSs relying on network-based methods and three IDSs relying on both host-based and network-based methods. However, the IDS model used in those studies was out of date and depended on the local machine more than on the network traffic during the analysis stage.

In 2013, Ganapathy et al. [46] presented a survey on intelligent techniques for feature selection and classification-based intrusion detection in networks. This survey considered fuzzy techniques, neural networks, genetic algorithms, neuro-genetic algorithms, particle swarm intelligence and rough sets for Internet security protection and QoS enhancement. Moreover, these authors proposed new feature selection and classification algorithms. In their experiment, they used 19 flow-based features comprising basic features, packet content features and traffic features.

In 2014, Mitchell and Chen [47] surveyed 60 papers on IDSs designed for wireless environments. Their survey revealed the strengths and weaknesses of IDS techniques for wireless local area networks (WLANs), wireless mesh networks (WMNs), wireless personal area networks (WPANs), wireless sensor networks (WSNs), cyber-physical systems (CPSs), ad hoc networks and mobile telephony.

Mitchell and Chen [47] proved that an anomaly-based IDS is the most suitable design for mobile telephony systems. However, such IDSs face challenges in terms of their high false positive rate and computational complexity [47]. High false positive and false negative rates reduce the QoS of a mobile network system. If any user packet is dropped by mistake, the user will suffer a billing error, and the user packet will be delayed [47]. Anomaly-based IDSs also face challenges with regard to illegal analysis methods, such as packet-based methods, that infringe on user privacy [47]. This survey proposed the detection latency as a critical metric for use in future research.

Also in 2014, Butun et al. [48] surveyed 18 papers focusing on mobile ad hoc networks (MANETs) and 17 papers focusing on WSNs in their survey on IDSs in WSNs. These authors discussed the feasibility of using systems designed for MANETs in WSNs. The possible security attacks against WSNs were divided into two categories: passive attacks and active attacks. These authors proved that IDSs are very important for the security of WSNs and that an IDS designed for a WSN must have certain special characteristics, including low power consumption. A WSN is a resource-constrained environment, so the effectiveness of an IDS in a WSN depends on its effect on the energy consumption of the network.

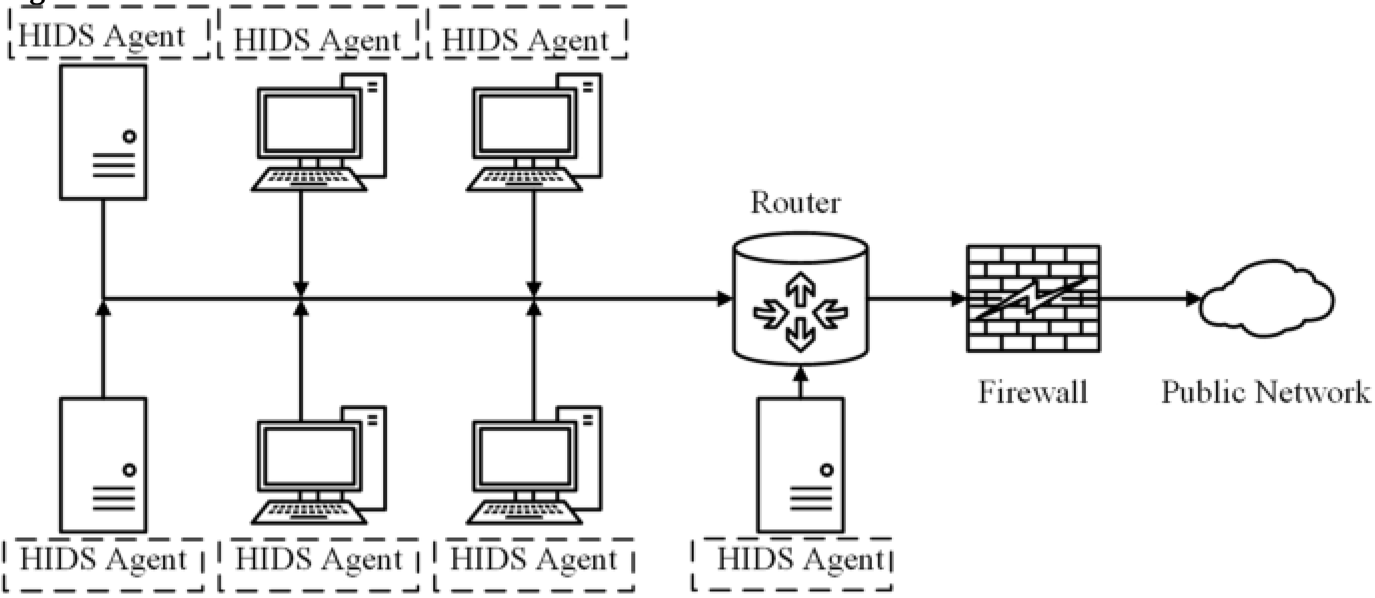
Butun et al. [48] recommended the use of a hierarchical IDS model to solve the energy consumption issue in WSNs. In accordance with the relevant application requirements, Butun et al. [48] recommended using a distributed IDS scheme for mobile applications, a centralized IDS scheme for stationary applications and a hierarchical IDS scheme for cluster-based applications.

IDSs: types and methods

The implementation of an IDS depends on the environment. A host-based intrusion detection system (HIDS) is designed to be implemented on a single system and to protect that system from intrusions or malicious attacks that will harm its operating system or data [49].

A HIDS generally depends on metrics in the host environment, such as the log files in a computer system [50]. These metrics or features are used as input to the decision engine of the HIDS. Thus, feature extraction from the host environment serves as the basis for any HIDS. The operational structure of a HIDS and its location in the network are shown in Fig. 6.

Fig. 6

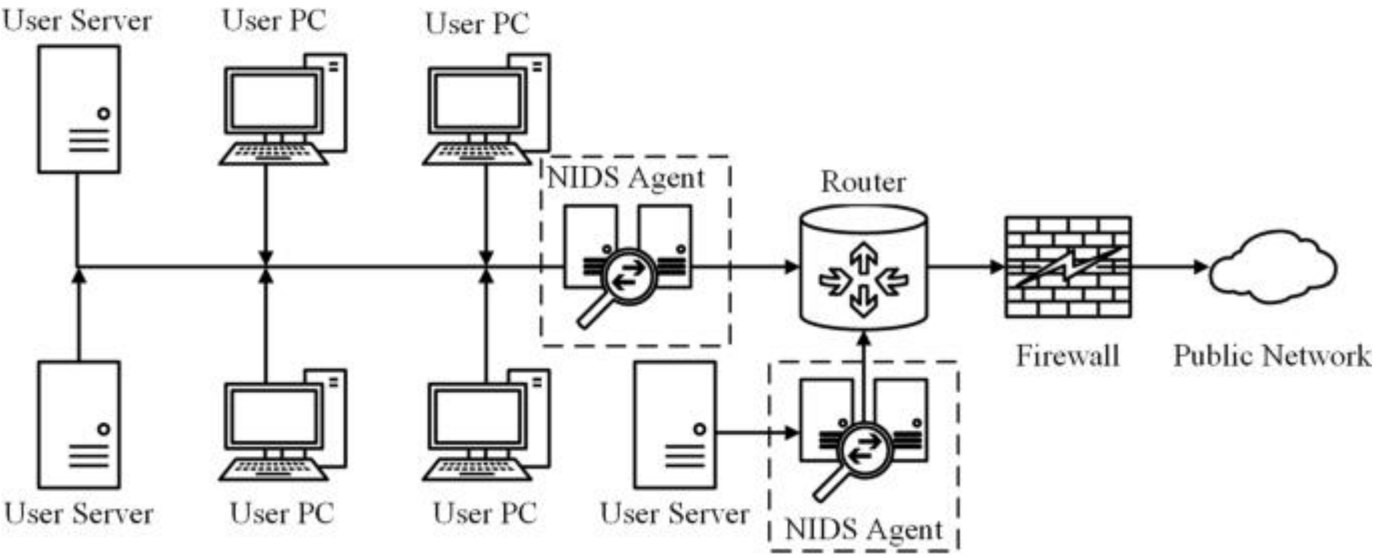


Generic architecture of a host-based IDS (HIDS). The operational structure of a HIDS and its location in the network

Full size image

A network-based intrusion detection system (NIDS) sniffs network traffic packets to detect intrusions and malicious attacks [50]. A NIDS can be either a software-based system or a hardware-based system. For example, Snort NIDS is a software-based NIDS [51]. The operational structure of a NIDS and its location in the network are shown in Fig. 7.

Fig. 7



Generic architecture of a network-based IDS (NIDS). The operational structure of a NIDS and its location in the network

Full size image

Network expansion and increasing traffic volumes necessitate the implementation of IDSs as hardware systems, such as a smart sensor architecture [52]. For example, field programmable gate arrays (FPGAs) can be used as the basis of a hardware-based NIDS. The special characteristics of FPGAs, such as their ability to support high-speed interfaces, dynamic reprogramming and very high-volume data processing, make FPGAs very suitable for use in NIDSs [53].

IDSs: detection techniques

An IDS depends on algorithms for implementing the various stages of intrusion detection. There are a vast number of algorithms for all IDS types and methods. Some of these IDS algorithms will be discussed briefly in the section titled ‘IDSs Designed for IoT Systems’.

Additionally, some of these IDS algorithms can be used for multiple different detection techniques. Thus, this section focuses on lightweight anomaly-based IDS algorithms that can be used in IoT-based environments depending on the complexity, execution time and detection time requirements. Principal component analysis (PCA) is a lightweight algorithm that can be used for various detection techniques in IDSs; thus, the PCA algorithm will be discussed as a representative example in the following.

Mori et al. [54] have stated that “principal component analysis (PCA) is a commonly used descriptive multivariate method for handling quantitative data and can be extended to deal with mixed measurement level data.” Thus, PCA has been widely applied in various fields. As described by [55], PCA generates a set of variables depending on the variance-covariance structure of the original variables. These new variables are linear combinations of the original variables and are fewer in number than the original variables.

In IDSs, PCA is used as a dimensionality reduction and detection technique. Elrawy et al. [56] used the PCA approach to create an anomaly-based statistical and data mining IDS that depends on the division of the principal components into the most and least significant principal components. In this system, the detection stage relies on the major principal component score and the minor principal component score. In addition, PCA has been used in intrusion detection techniques based on payload modeling, statistical modeling, data mining and machine learning [56–58].

Misuse-based intrusion detection

A misuse-based intrusion detection technique uses a database of known signatures and patterns of malicious codes and intrusions to detect well-known attacks [59]. Network packet overload, the high cost of signature matching, and the large number of false alarms are three disadvantages of misuse-based IDSs [60]. In addition, the severe memory constraints in some types of networks, such as WSNs, result in low performance of misuse-based IDSs because of their need to store a large database of attack signatures [61].

Moreover, the signature and pattern databases in signature-based IDSs and pattern-matching IDSs need to be continuously updated. Such misuse-based IDSs are designed to detect malicious attacks and intrusions based on previous knowledge.

Anomaly-based intrusion detection

In an anomaly-based intrusion detection technique, a normal data pattern is created based on data from normal users and is then compared against current data patterns in an online manner to detect anomalies [62]. Such anomalies arise due to noise or other phenomena that have some probability of being created by hacking tools.

Thus, anomalies are unusual behaviors caused by intruders that leave footprints in the computing environment [63]. These footprints are detected in order to identify attacks, particularly unknown attacks.

An anomaly-based IDS operates by creating a model of the normal behavior in the computing environment, which is continuously updated, based on data from normal users and using this model to detect any deviation from normal behavior [64]. The advantages and disadvantages of various anomaly-based intrusion detection techniques are shown in Table 1. These techniques will be discussed in the following.

Table 1 A short comparison between anomaly IDS techniques with a focus on the advantages and disadvantages of each technique

Full size table

- A data mining approach is a means for extracting knowledge from a large amount of data, analogous to extracting gold from numerous rocks and sand [65]. The extracted knowledge is defined as interesting patterns in the data [66]. Such a pattern can describe the behavior of data from users or networks in a computing environment. The ability to automatically generate models that depend on the traffic description is one of the advantages of the data mining approach. Moreover, this approach can be applied in generalized IDSs and in any computing environment [67]. The data mining approach works perfectly for an online data stream that is unbounded, continuous and rapidly increasing in volume [68]. A procedure consisting of a rule learning stage, a clustering stage, a classification stage, and a regression stage is applied in the design of an IDS based on this approach [68].

- Machine learning is a technique that depends on two stages: the training or learning stage and the detection or testing stage [69]. The training stage depends on mathematical algorithms or functions that use normal data as a reference input to learn the characteristics of the computing environment. Then, in the detection stage, these characteristics are used for detection and classification [70]. Supervised learning is one type of machine learning technique in which the characteristics of the training dataset are used in the learning phase to create a classification model, which is then used to classify new unseen instances [71]. Unsupervised learning is a type of machine learning technique that depends on the features of the data without using clustered training data [71].

A pattern classification method in machine learning depends on pattern recognition, whereas a single classifier method depends on a single machine learning algorithm [72, 73].

- The statistical model approach depends on statistical mathematical operations [74]. The statistics of historical user behavior are used to create a normal model, and any deviations from this model are then detected. These deviations are considered abnormal data. The statistical model approach uses statistical mathematical operations applied to a training dataset to detect abnormal traffic from the observed traffic patterns [75].
- The rule model approach depends on the creation of rules for the computing environment. These rules are extracted from data traffic patterns. A rule-model-based IDS detects any anomalous data traffic that breaks these rules and considers any such anomaly as an attack [76]. The rule creation process depends on the historical system behavior. Thus, the system must be monitored for a long time to avoid an excessively high false positive rate.
- The payload model approach depends on the packet traffic of a specific port or user for a given application. In a signature-based IDS, the payload model is based on pattern matching to identify attack packets with specific characteristics [77]. By contrast, an anomaly-based IDS that uses the payload model approach creates a model that depends on bytes or calculations from bytes that describe the normal characteristics of the packet payload [57].
- The protocol model approach depends on monitoring protocols in different layers of the computing environment. An IDS based on this approach detects anomalies associated with a specific protocol or a protocol that is not present in the normal model. A specification-based approach, a parser-based approach or an approach based on application protocol keywords can be used to analyze the protocols in a computing environment [78].
- The signal processing model approach depends on traffic analysis using signal processing methods. An IDS based on this approach creates a normal pattern by capturing the statistics of normal data traffic and the data distribution over time, and any deviation from this pattern is considered to be an anomaly [79].

Specification-based intrusion detection

The concept of a specification-based IDS was proposed by Ko et al. [80] in 1997. They proposed a monitoring and detection system based on security specifications that determine the normal behavior of the system to be protected. These security specifications are created based on the functions and security policies for this system. Thus, operating sequences that are not included in the system behavior are considered security violations [81].

The most important challenge in designing a robust specification-based IDS is creating a formalism that captures the valid operating sequences of the system. Therefore, the cost of defining the specification “trace policy” and the difficulty of evaluating and verifying the specifications limit the real-world applicability of specification-based IDSs. A specification-based IDS learns the root characteristics of attacks and detects known attacks like a misuse-based IDS, and it also has the ability of anomaly-based IDSs to detect unknown attacks, such as operating sequences that are not included in the normal behavior of the system [82–84].

IDSs: performance evaluation

The measures used to assess IDS performance depend on four factors, namely, the numbers of true positives (*α*), true negatives (*δ*), false positives (*γ*) and false negatives (*β*), as described in Table 2. Following [85, 86], these factors and the performance metrics for IDSs are described below.

When predicting the anomaly class, a true positive (α_A) is a correct classification that indicates an intrusion. A true negative (δ_A) is a correct classification that indicates no intrusion. A false positive (γ_A) is an incorrect classification that indicates an intrusion when there is no intrusion. A false negative (β_A) is an incorrect classification that indicates no intrusion when there is an intrusion. The true positive rate (TPR), which describes the probability of detecting intrusions, is calculated as:

$$TPR=\alpha_{A}\alpha_{A}+\beta_{A}TPR=\alpha A\alpha A+\beta A$$

(1)

The false positive rate (FPR), which describes the probability of incorrectly identifying normal behavior as an intrusion, is calculated as:

$$FPR=\gamma_{A}\gamma_{A}+\delta_{A}FPR=\gamma A\gamma A+\delta A$$

(2)

The recall (R), which describes the percentage of the total relevant records in a database that are retrieved by searching, is calculated in the same way as the TPR. The precision (P), which describes the percentage of relevant records among the records retrieved, is calculated as:

$$P=\alpha_{A}\alpha_{A}+\gamma_{A}P=\alpha A\alpha A+\gamma A$$

(3)

The F-score (F), which describes the balance between P and R, is calculated as:

$$F=2\ast P\ast RP+RF=2\ast P\ast RP+R \tag{4}$$

The overall success rate, which describes the percentage of correct classifications, is calculated as:

$$SuccessRate=\alpha_A+\delta_A\alpha_A+\delta_A+\gamma_A+\beta_A SuccessRate=\alpha A+\delta A\alpha A+\delta A+\gamma A+\beta A \tag{5}$$

$$ErrorRate=1-SuccessRateErrorRate=1-SuccessRate \tag{6}$$

When predicting the normal class, the same definitions and equations can be used, except with the parameters α_N, β_N, γ_N and δ_N.

IDSs designed for IoT systems

Due to the security challenges facing IoT systems, methods that can proactively identify new attacks are most suitable for protecting IoT networks. Thus, a robust IDS that can detect new attacks in IoT-based smart environments is required. An overview of the IDSs that have been proposed for IoT systems is shown in Table 3.

According to the recommendations in recent surveys of IDSs for IoT systems [3, 7], this paper focuses on the features of all IDS methods for the IoT that can be applied in smart environments. IoT systems require special security measures with particular characteristics that are not offered by traditional IDSs.

Liu et al. [87] proposed an artificial immune IDS for IoT networks. This system can adapt to the IoT environment and automatically learn new attacks. The system is based on machine learning and a signature-based model. The adopted machine learning approach is designed after the mechanisms of artificial immune systems. The objective of the system is to increase the security of the IoT network; thus, it is a network IDS. This system has two main features: self-adaptation to new environments and self-learning of new attacks.

Kasinathan et al. [88] proposed an IDS that detects DoS attacks based on 6LoWPAN in IoT networks. They proposed a DoS detection architecture that consists of an IDS probe, a DoS protection manager and a Suricata IDS [89]. They designed this system based on a study of the vulnerabilities present in IP-based WSNs. The Suricata [89] IDS runs on a host computer; thus, the advantage of this system is that it can overcome the problem of power consumption, thus conserving power resources in WSNs.

Moreover, Kasinathan et al. [90] proposed an enhanced IDS for detecting DoS attacks based on 6LoWPAN in IoT networks. This system depends on the DoS detection architecture presented in [88]; its main new elements are a frequency agility manager (FAM) and security incident and event management system (SIEM). These elements together create a monitoring system that can monitor large networks.

Jun and Chi [91] proposed an IDS integrated with complex event-processing (CEP) technology. The benefit of CEP technology is the ability to identify complex patterns via real-time data processing. The event-processing IDS architecture consists of an event filtering unit, an event database unit, a CEP unit and an action engine unit. The system depends on an event-processing model that uses the rule model approach to detect intrusions.

The main features of this system are that it operates in real time and shows high performance in detecting intrusions in an IoT system using an event-processing mechanism.

Krimmling and Peter [92] proposed a NIDS that depends on machine learning for anomaly-based and signature-based intrusion detection. The system framework is designed for smart public transport applications that use CoAP. The main features of this system are its applicability to CoAP applications and its reliance on a lightweight algorithm.

Butun et al. [93] proposed a NIDS for WSNs that combines the statistical model approach and the rule model approach. The system is based on a downward-IDS and an upward-IDS in accordance with the hierarchical WSN structure. The downward-IDS detects abnormal behavior of the member nodes, and the upward-IDS detects abnormal behavior of the cluster heads. The main features of this system are its applicability to hierarchical WSNs and its dependence on WSN clustering.

Surendar and Umamakeswari [82] proposed a constraint-based specification IDS for IoT networks using 6LoWPAN. This system maintains efficiency in terms of QoS metrics while detecting sinkhole attacks. The system isolates malicious nodes and reconstructs the network without these nodes. This IDS is a specification-based IDS that depends on behavioral rules and uses the protocol model approach.

The main features of this system are that it detects sinkhole attacks, preserves QoS and isolates malicious nodes.

In addition, Le et al. [83] proposed a specification-based IDS for IoT networks using 6LoWPAN for the detection of several topology attacks against the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL), such as sinkhole, rank, local repair, neighbor, and destination oriented directed acyclic graph (DODAG) information solicitation (DIS) attacks. In a DIS attack, for example, the attacker increases the overhead in the network by using malicious nodes to send DIS messages [94] with fake IP addresses to the malicious nodes' neighbors, forcing these other nodes to generate DODAG information objective (DIO) messages [83, 94]. The proposed IDS depends on analyzing the protocol behavior from trace files to learn the route establishment and maintenance procedures for a stable topology. The main features of this system are its high efficiency in detecting RPL topology attacks in an energy-efficient manner and its applicability to large-scale networks.

Moreover, Bostani and Sheikhan [84] proposed a hybrid IDS for IoT networks using 6LoWPAN for the detection of several RPL attacks. This system depends on specification-based intrusion detection modules, serving as IDS agents, in the router nodes and an anomaly-based intrusion detection module, serving as the main IDS, in the root node. The main features of this system are a reduction in the number of communication messages due to the lack of additional control messages or monitor nodes in the IDS design and its applicability to large-scale networks.

Garcia-Font et al. [95] proposed a NIDS for WSNs that depends on a machine learning approach and a signature model. They used a signature-based detection engine and an anomaly-based detection engine to improve the detection rate and the FPR. The system is designed to help smart city administrators detect intrusions using the IDS and an attack classification schema. The objective of the system is to detect intrusions in WSNs in different smart city environments. The main feature of this system is its applicability to large-scale WSNs.

Fu et al. [96] proposed a NIDS that depends on signature-based and protocol-based anomaly detection. The proposed IDS structure focuses on detecting attacks on IoT networks without being affected by the heterogeneity of such a network. The detection method depends on comparing the abstracted action flows in the data packets against three databases based on the protocol type information for each packet. These databases are a standard protocol library, an abnormal action library and a normal action library. The proposed approach consists of an event monitor, an event database, an event analyzer, and a response unit. This approach provides a uniform intrusion detection method for IoT networks based on automata theory. The main features of this system are the classification of attacks into three categories and the development of graphical user interface (GUI) tools to graphically present the abstract action flows and detect possible intrusions.

Deng et al. [97] proposed a NIDS that depends on the fuzzy c-means clustering (FCM) algorithm and the PCA algorithm. The system combines a machine learning approach with a data mining approach to improve the accuracy of intrusion detection for IoT networks. The PCA algorithm is used for feature selection and reduction. The FCM algorithm is used as a clustering method. The KDD-CUP99 dataset [98] was used to evaluate the proposed system. The main features of this system are its light weight and its ability to improve the detection efficiency by achieving a low FPR.

Amouri et al. [99] proposed a NIDS based on the protocol model approach and machine learning. This system consists of two detection stages. In the first stage, namely, local detection, network behavior data are collected by dedicated sniffers to generate a set of correctly classified instances (CCIs) using a supervised learning approach based on decision trees. In the second stage, namely, global detection, CCIs are collected by supernodes to generate time-based profiles called the accumulated measures of fluctuation (AMoFs) for malicious and normal nodes separately.

The main features of this system are its low computational complexity and low resource requirements.

Liu et al. [100] proposed a NIDS that depends on the suppressed fuzzy clustering (SFC) algorithm and the PCA algorithm. This system combines machine learning and data mining to improve the accuracy of intrusion detection in high-dimensional space. The PCA algorithm is used for feature extraction. A novel prejudgment-based intrusion detection method using PCA and SFC is applied that divides the dimension-reduced data into high-risk and low-risk data. The main feature of this system is its adaptability to high-dimensional spaces, such as the IoT space. Moreover, the efficiency and effectiveness of the IDS are optimized by reducing the detection time and increasing the accuracy by means of a frequency self-adjustment algorithm.

Abhishek et al. [101] proposed a NIDS that depends on the packet drop probability (PDP) in an IoT device to monitor gateways and detect malicious gateways. The system uses the statistical model approach for anomaly-based intrusion detection using a likelihood ratio test to detect malicious gateways, which corrupt the communications between IoT devices and access points. One of the disadvantages of this system is that it can only detect malicious getaways that affect downlink packets; it does not consider malicious getaways that affect uplink packets from IoT devices. The main features of this system are that it is based on theoretical foundations instead of requiring training data and that it can detect malicious gateways in real time.

Oh et al. [102] proposed a lightweight malicious-pattern-matching IDS. They stressed that traditional IDSs are not applicable for smart objects due to the limited memory size and battery life of these objects. Thus, a powerful and lightweight IDS is required because of these restrictions. Oh et al. proposed the auxiliary skipping (AS) algorithm, the early decision with boundary searching (EBS) algorithm, and an approach that uses both AS and EBS (AS-EBS).

These algorithms reduce the number of matching operations that must be performed [102]. The system depends on a pattern-matching approach based on signature detection. The advantage of this system is that it can be applied to smart objects with limited memory size and battery life. The main features of this system are the reduced memory size required for matching operations, the reduced workload for processing on smart objects, the increased speed of processing, and its scalable performance for a large number of patterns.

Summerville et al. [103] proposed an ultralightweight deep packet anomaly detection approach that can be implemented on small IoT devices. The system is designed with a dependence on the bitwise AND operation and uses a payload model approach for anomaly-based intrusion detection. The main features of this system are its low latency, high throughput and ultralight weight.

Mohan et al. [37] proposed a HIDS that depends on signature-based and rule-based anomaly detection. The system uses the traditional signature-based technique in combination with Snort-rule-based intrusion detection. Thus, the system can detect known attacks using the signature database and unknown attacks using SNORT rules. The main challenge of this system is privacy because the system uses a deep packet inspection technique to detect attacks. The main features of this system are its simplicity and self-learning capability.

Arrignton et al. [104] proposed a HIDS that depends on a machine learning approach for anomaly-based intrusion detection. The machine learning approach is based on the mechanisms of artificial immune systems. The main features of this system are its use of a behavioral modeling IDS (BMIDS) to decide whether behavior is acceptable and its increased detection sensitivity achieved by canceling out environmental noise.

Gupta et al. [105] confirmed that the threat of attacks in IoT systems affects not only the computational environment but also human life and the economy. For this reason, they proposed a computational-intelligence-based IDS for wireless communications and IoT systems. They proposed a three-tier architecture as the basis of an intelligent IDS suitable for wireless networks; this architecture consists of an information storage unit, a computational intelligence and optimization unit, and a clustering and intrusion reporting unit. This system depends on a machine learning approach for anomaly-based intrusion detection.

The machine learning approach applied in this IDS is based on the swarm intelligence (SI) paradigm, which is a specific type of computational intelligence (CI) paradigm [105]. The system targets IP addresses to detect attacks; thus, it has the disadvantage that it cannot be applied in regions of WSNs that do not use the TCP/IP protocol. The main feature of this system is its ability to operate as both a NIDS and a HIDS.

Raza et al. [106] proposed a hybrid-based IDS for IoT networks using 6LoWPAN for the detection of several RPL attacks. They proposed the SVELTE IDS, which consists of a 6LoWPAN mapper unit, an intrusion detection unit and a mini-firewall unit. The 6LoWPAN mapper unit collects information about the RPL network. The intrusion detection unit analyzes the data from the 6LoWPAN mapper unit to detect intrusions. The mini-firewall unit filters unwanted traffic. This system is designed for distributed and centralized IDS placement strategies. The main features of this system are its light weight and energy efficiency.

Khan and Herrmann [107] proposed three algorithms based on the protocol model approach using a trust management mechanism for IoT networks. One of these algorithms is neighbor based trust dissemination (NBTD), which can be used to implement a NIDS in a border router using the centralized approach. The second algorithm is tree based trust dissemination (TTD), which can be used to implement a HIDS in a small network with extra-high communication costs using the distributed approach. The third algorithm is clustered neighbor based trust dissemination (CNTD), which can be used to implement a NIDS using the distributed approach to reduce the number of packet exchanges compared with the NBTD algorithm. The main features of these algorithms are their light weight, energy efficiency and applicability in healthcare environments.

Performance analysis

In this section, a descriptive statistical analysis is applied to the reviewed papers based on several performance metrics: the TPR, FPR, energy consumption, processing time and performance overhead. The suitability of an IDS for IoT-based smart environments depends on these metrics; thus, they are the focus of this study.

In a traditional communication network, the performance of an IDS depends on the TPR and FPR only. In an IoT-based smart environment, the energy consumption, processing time and performance overhead of an IDS are also of critical interest. Because of the power and memory limitations of IoT devices, these metrics are very important to the QoS of an IoT system. Therefore, they are important performance metrics for an IDS designed for IoT-based smart environments.

Table 4 summarizes the technical details of the surveyed papers in terms of the important performance metrics of the proposed IDSs and their effects on smart environments. In Table 4, the symbol - indicates that no experimental result is available for the corresponding metric. The symbol * indicates that no numerical result was determined for this metric. The terms MAX and MAX range refer to the maximum result or maximum range result, respectively, for this metric.

Table 4 A descriptive statistical analysis of the surveyed IDSs in terms of the important performance metrics affecting IDS suitability in IoT systems

Full size table

Table 4 merely summarizes the experimental results from the surveyed papers; it is not intended as a comparison of these results. The experiments reported in these papers were performed under different conditions and using databases of different sizes. Thus, a standard IoT workbench with a standard IoT database would be required to conduct a fair comparison.

Tables 3 and 4 illustrate that the majority of researchers focus on three parameters during the testing stage: the TPR, FPR and processing time. Thus, IDSs are designed based on four features. First, such a system should be compatible with IoT systems. Second, it should be able to detect attacks in real time. Third, it should depend on lightweight algorithms. Fourth, it should be scalable.

Tables 3 and 4 show that some previous IDS studies did not present experimental results, whereas others presented only system methodologies that were not subjected to real experiments. Only a few previous IDS studies have presented practical results for the performance metrics that characterize the suitability of IDSs for IoT-based smart environments. Thus, the development of IDSs for IoT-based smart environments is still in an incipient phase.

Discussion and future outlook

Integrity, confidentiality, and availability are three important factors in IoT systems. In most cases, applications that use the IoT model are considered to be vital, such as industrial and medical applications. On the one hand, these applications can be real-time applications; thus, network delay and latency directly affect their performance. On the other hand, attacks such as DoS, DDoS, probing, and RPL attacks can degrade the usability of these applications. Thus, security issues can be considered a life-threatening concern in e-health systems, for example [108]. Consequently, powerful security measures are required in IoT networks. Such a security mechanism must protect the IoT network and its resources without impacting the system's performance or user privacy.

Moreover, IoT-based smart environments consist of a wide range of devices, sensors and IoT objects from different vendors and based on different IoT platforms. Thus, interoperability issues prevent the emergence of IoT technology at a large scale [109]. Interoperability and standardization issues must be considered in designing IDSs for IoT-based smart environments.

IoT networks suffer from power efficiency problems; thus, a lightweight IDS that requires only a small number of computational operations is needed. In a HIDS, the IoT devices must simultaneously perform the necessary computational operations for the IDS and for IoT services. Thus, power resources and battery life must be considered in HIDS designs. Because of the power and memory limitations of IoT systems, the energy consumption, processing time and performance overhead of an IDS are important performance metrics. Thus, these metrics must be considered when designing IDSs for IoT-based smart environments. These issues should receive greater focus in research on HIDSs for such environments.

Privacy is another important factor in IoT systems. Deep packet inspection techniques are considered a violation of privacy. Such techniques and other techniques with similar characteristics are therefore undesirable. Moreover, the blocking of normal data packets affects IoT applications and services. This effect is very harmful, particularly for vital and real-time applications, such as industrial and medical applications. Therefore, introducing a smart system without deep packet inspection requires trusting that the operations in the IoT system will prevent any unauthorized access to IoT objects, thus helping to solve the user privacy problem. A new IDS design with a very low FPR and a very high detection accuracy is required for application in vital and real-time applications because traditional IDSs cannot satisfy these requirements.

An IDS based on a hybrid intrusion detection technique is required to detect different types of attacks from different computational environments. The IDS must be compatible with the 6LoWPAN protocol to detect attacks in WSNs in IoT networks. Furthermore, an autonomous IDS that can detect intrusions without human intervention is required for application in the IoT environment.

IDS placement is also a serious issue that must be considered when designing any type of IDS, whether it is a NIDS or a HIDS. The placement of the IDS in the IoT network will affect the overall efficiency of the IDS. There are two general IDS placement strategies: centralized and distributed. The centralized strategy offers the advantage of centralized management but can also lead to system processing overload, which may affect the QoS in IoT networks. The distributed strategy has the advantages of reducing the amount of monitored traffic and increasing the processing capacity. However, implementing an IDS in different regions of an IoT network is a challenge due to the associated management issues.

Finally, there is a need for both normal and anomaly databases that are up to date and integrated with IoT applications and services. These databases will be very useful for testing different IDS types and techniques in IoT environments. The ability to perform successful and meaningful IDS comparisons will depend on these databases.

Concluding remarks

As the numbers of IoT users, services, and applications increase, an urgent need for a robust and lightweight security solution that is suitable for use in IoT environments is emerging. Furthermore, IoT networks are the basis of smart environments; thus, any deficiencies in the security of these IoT networks will directly influence the smart environments on which they are based. Attacks such as DoS, DDoS, probing, and RPL attacks affect the services and applications offered in IoT-based smart environments; thus, the security of IoT environments is a very serious issue. An IDS is one possible solution for this issue. This has paper presented a survey of IDSs designed for IoT environments. Recommendations for designing a robust and lightweight IDS were also discussed.

In this survey, several papers were investigated. These papers mainly study the design and implementation of IDSs for use in the IoT paradigm that can be applied in smart environments. The features of all IDS methods presented in these papers were summarized. Moreover, this paper proposed some recommendations that must be considered when designing an IDS for the IoT, such as the need for a powerful and lightweight system with a suitable placement strategy that does not adversely affect the integrity, confidentiality, and availability of the IoT environment. This study showed that there is a need to design an integrated IDS that can be applied in IoT-based smart environments. This design will need to be tested on a unified IoT database. The question of the placement strategy must be considered in this design.

Future work will investigate the design of a high-performance hybrid IDS specifically designed for IoT-based smart environments based on the recommendations of this study. Moreover, the security vulnerabilities of IoT enabling technologies and protocols will be considered in the IDS design. In addition, the IDS will be implemented on programmable reconfigurable hardware devices, such as FPGAs, to facilitate adaptation to IoT-based smart environments. The design should be suitable for both distributed and centralized placement strategies and have the ability to detect different types of attacks.