INNOVATION IN SCIENCE AND TECHNOLOGY

ISSN 2788-7030 JUN. 2026 VOL. 5, NO. 2

CONTENTS

Paradigm Academic Press
Innovation in Science and Technology
ISSN 2788-7030
JUN. 2026 VOL.5, NO.2

Hacking Is an Unauthorized Access in Electronic Devices: Skills Technologies Are Necessary to Enhance Defense Strategies

Haradhan Kumar Mohajan[1]

[1] Chairman and Associate Professor, Department of Mathematics, Premier University, Chittagong, Bangladesh

Correspondence: Haradhan Kumar Mohajan, Chairman and Associate Professor, Department of Mathematics, Premier University, Chittagong, Bangladesh.

doi:10.63593/IST.2788-7030.2026.06.001

Abstract

Hacking is the technique of finding the weaknesses in the network system that exploits to gain unauthorized access to personal or business data that are in the networks, and the hacker is responsible for the legal consequences of his/her actions. The hacker has basic knowledge, desire, motivation, deep patience, planning workability, and financial supports. Usually an unethical hacker (black hat hacker) is a malicious guy who tries to steal, leak, and destroy confidential and valuable data and other sensitive information of the computer systems without permission of the user. S/he can usually be organized into two types of attacks: mass attacks and targeted attacks. On the other hand, an ethical hacker (white hat hacker) tries to strengthen the security mechanisms of the organization by exploring the weaknesses of it. This study tries to discuss the aspects of unethical hacking, types of hackers, and their behaviors.

Keywords: hacking, hacking techniques, hacker, cybercrime, cyber security

1. Introduction

After the advent of internet anybody can communicate with any person in any part of the world and makes any deal with just a click with any electronic device connected to network. Consequently, the physical distance barrier has been removed and transaction time has been highly reduced, but the cheating and frauds become very high that increase global concern (Begum et al., 2016). Information security is the protection of information system from unauthorized access, use, disclosure, disruption, modification, and destruction. In the digital world in our daily lives, we have observed that security attacks are growing in an exponential manner (Pal, 2016). Nowadays most of the currently available websites, networks, and applications are poorly and hastily configured, and malicious hackers can exploit these vulnerabilities and security gaps (Yaacoub et al., 2021).

Hacking is the act of identifying and exploiting weaknesses in digital devices, such as computers, smartphones, tablets, and networks by an unauthorized source through the installment of dangerous malware (Oakley, 2019). It is not a basic or regular activity that consists of two paths. The first path is towards the legal and permissioned work while the other path is towards the illegal and unapproved work (Memon et al., 2020). It is the most common form of cybercrime that can be used from monetary gain to political interest. It may be in different forms, such as web-spoofing, email bombing, Trojan attacks, phishing, fake websites, spyware, electronic bulletin boards, information brokers, virus attacks, wormhole attack, password cracking, etc. (Fuchs, 2014; Mohajan, 2025b).

Hackers are a type of programmers and software engineers, who usually steal the secret data, such as phone numbers, credit card details, addresses, online banking passwords, etc. (Kumar & Agarwal, 2018). They are mostly nerd, wild, and teenager types, and spend an average time of 57 hours a week on their system. They are experienced in C, C++, and Perl programming languages, and familiar with the UNIX (Chirillo, 2002; Mohajan,

2025c).

2. Literature Review

A literature review is a comprehensive summary and critical evaluation of existing scholarly work on a specific topic that is a section of a larger work of an article, a conference paper, a book, a book chapter, or a thesis (Creswell, 2013). It is an overview of previously published works on a particular topic. It involves finding scholarly literature on the topic, reading and taking notes, analyzing or evaluating the literature, writing the literature review, etc. (Torraco, 2016). It highlights how knowledge has evolved within the field, what has already been done, what is generally accepted, what is emerging, and what is the current state of thinking on the topic (Galvan, 2015). It provides the current state of knowledge, identify gaps in the research, and establish a foundation for new research. A good literature review has a proper research question, a proper theoretical framework, and a chosen research methodology (Baglione, 2012).

S. K. Ashwini and K. Thippeswamy have provided the brief information on ethical hacking, ethical hacking types, how ethical hacking works, methodology and hacking tools, and some needs and limitations of the ethical hacking (Ashwini & Thippeswamy, 2018). Rajinedr Singh and Shakti Kumar have shown that in Information and Communications Technology (ICT) no one is isolated, and are connected to each other through internet and new networking technology. They have observed that the network and windows security can be simple or complex depending upon the requirements. They have analyzed the security threats and internet protocol to determine the necessary security technology through the understanding of its importance, history, and various technologies that can be used to provide security measures (Singh & Kumar, 2014).

Imran Memon and his coauthors have shown that hacking is the most genius field of computer science that is getting bigger with the passage of time. They have used the methods and techniques of hacking, types of hacking, types of hackers, types of hacking attacks, common tools of hacking, the geography of hackers, phases of hacking, protection processes, past reports, instruments of hacking, and at last the future discussion on this field (Memon et al., 2020). Priyank Dinesh Gada has wanted to show how credit card hacking works, how smartphones can be hacked, how SIM swapping works, and other techniques with a practical approach. He also shows how to change International Mobile Equipment Identity (IMEI) number and other black hat hacking techniques used by hackers (Gada, 2024).

Md. Shaiful Islam and his coworkers have wanted to examine the challenges of cyber security to service delivery in the public administration. They have identified the cyber security issues, cyber threats, vulnerabilities, awareness among employees, and steps taken to ensure cyber security in the offices of public administration in Bangladesh (Islam et al., 2023). Minakshi Bhardwaj and G.P. Singh have discussed three types of attacks against computer systems: physical, syntactic and semantic attacks. A physical attack uses conventional weapons, such as bombs or fire. A syntactic attack uses virus-type software to disrupt or damage a computer system or network. A semantic attack is a more subtle approach. Its goal is to attack users' confidence by causing a computer system to produce errors and unpredictable results (Bhardwaj & Singh, 2011).

3. Research Methodology of the Study

Research is a creative and systematic work undertaken to increase the stock of knowledge that involves the collection, organization, and analysis of evidence to increase understanding of a topic; characterized by a particular attentiveness to control sources of bias and error (Kara, 2012). It searches for knowledge and truth that must be controlled, rigorous, systematic, valid and verifiable, empirical, and critical (Grover, 2015). Methodology is the analysis of the principles of methods, rules, and postulates employed by a discipline (John, 2017; Mohajan, 2018b). Research methodology is the systematic plan for conducting a study, detailing the specific procedures and techniques used to collect, analyze, and interpret data to answer research questions (Groh, 2018). It includes the overall research design, data collection methods, and data analysis techniques, explaining why these choices are made to ensure the results are valid, reliable, and replicable. It is a science of studying how research is to be carried out in a systematic way to solve a problem (Creswell, 2013). In this study we have dependent on the secondary data sources that are collected from research articles, hand books, books, thesis, and project reports (Mohajan, 2020). We have also stressed on the reliability and validity for the acceptance of ethical values (Mohajan, 2017). At the starting we have highlighted on the overview of hacking, and then we have discussed on hacker and its types. At the third step we have briefly discussed the tools of hacking, types of hacking, and finally we have highlighted on steps of hacking.

4. Objective of the Study

Hacking is the process of attempting to gain unauthorized access to computer resources. It has been a part of computing for last 50 years (Pal, 2016). There are many types of hacking in networking system, such as website hacking, network hacking, ethical hacking, unethical hacking, email hacking, password hacking, computer hacking, bank account hacking, etc. (Cekerevac et al., 2018). In the 21st century, hacking becomes one of the

most important skills in the networking system that is getting bigger with the passage of time (Mohajan, 2025c). Regular auditing, vigilant intrusion detection, good system administration practice, and computer security awareness are all essential parts of security efforts of an organization (Oakley, 2019). The hacker is a person who has a basic knowledge, desire, motivation, and some money. S/he must have a large dose of patience and planning workability. However, neither all hackers are all the same, nor all hackers have the same goals (Gada, 2024). Main objective of this study is to discuss the background and overview of hacking with identification and classification of hackers (Mohajan, 2025a). Some other minor objectives of the study are as follows (Mohajan, 2018a):

1) to highlight on hacker and its types,
2) to focus on tools and types of hacking, and
3) to emphasize on steps of hacking.

5. Overview of Hacking

Hacking is the act of compromising digital devices and networks through unauthorized access to an account or computer system, and a hacker is responsible for the legal consequences of his actions. It refers to the misuse of devices, such as computers, smartphones, tablets, and networks to cause damage to corrupt systems, gather information on users, steal data and documents, and disrupt data-related activity (Cekerevac et al., 2018). They also try to collect various data, such as business and private data, bank accounts, emails, websites, databases, etc. to damage and destruct them or for personal gaining (Sheikh, 2021). Hacking is not a regular activity that consists of two paths: The first path is towards the legal and permissioned work, and it is called ethical hacking while the other path is towards the illegal and unapproved work, and it is called unethical hacking (Memon et al., 2020; Mohajan, 2025d).

6. Hacker and Its Types

A hacker is a person skilled in information technology that solves problems by non-standard means, and achieves success. He is an expert computer programmer who has huge exceptional knowledge of computer programming and has enough information on the systems to hack (Gregg, 2006). The term “hacker” (skillful programmer) was coined in the 1960s by a couple of Massachusetts Institute of Technology (MIT) students of Tech Model Railroading Club (TMRC) to describe the students who creatively solved technical problems, especially those who found clever programming shortcuts (Sterling, 1992). It largely referred to persons capable of implementing elegant and technically advanced solutions to technologically complex problems. Initially it was a positive term for experts; and later it was adopted by mainstream media to describe computer criminals (Kartalopoulos, 2008).

The hacker has knowledge in both hardware and software operations. Consequently, s/he is the most dangerous threat to the information systems. There are mainly three types of hackers: white hat, black hat, and gray hat hackers (Moore, 2005). Other four types of hackers based on what they hack and how they perform the hack are green hat, red hat, blue hat hackers, and script kiddie (Levy, 1984). They are professional hackers and highly technical people who perform hacking for earning their personal incomes. For example, many white hat hackers are employed worldwide to test organizations’ computer security systems (Hoffman, 2013). Data security is a huge problem in the modern world. At present worldwide security spending reaches to $100 billion (Shital, 2023).

6.1 White Hat Hackers

White hat hackers are known as the good guys or ethical hackers or sneakers who hack with permission from the data owner, and who have non-malicious nature (Levy, 1984). They are certified hackers who work for the Government, non-government organizations (NGOs), and private firms under the rules and regulations provided by the Government through performing penetration testing and identifying loopholes in their cyber security (Banda et al., 2019). They work ethically aiming to improve security, and learn hacking from courses. They use their skills to find vulnerabilities and help organizations to improve their defenses (Cornwall, 1986). They are considered good hackers, and work for the benefit of the network system. They use their abilities for good, ethical, and legal purposes rather than bad, unethical, and criminal purposes (Shanmugapriya, 2013). The International Council of Commerce Consultants (ICC) is one of the organizations that has developed certifications, courseware, classes, and online training covering the diverse arena of white hacking (Pal, 2016).

White hat hackers are computer security specialists and try to find loopholes in the protected networks or the computer systems of an organization and improve the security. Usually, they are either hired or contracted by organizations to evaluate their security parameters (Shital, 2023). They have hacking skills for defensive purposes, and locate weaknesses and implement countermeasures. They provide detail security vulnerability reports to the organization and inform it of how they have gained access, and allow the organization to improve

its defenses (Hoffman, 2013).

6.2 Black Hat Hackers

The term “black hat hacker” was coined by American free software movement activist and programmer Richard Stallman. They are also known as “crackers” who hack the systems with malicious intent, and exploit vulnerabilities for personal gain, financial profit, and cause harm (Moore, 2005). They are considered as bad hackers and target to destroy the system, compromise the privacy, block communication, damage the computer system, and thieve the confidential information. They are familiar with computer networks, Linux, cryptography, and other skills (Hoffman, 2013). They are highly skilled and have knowledge in various hacking techniques (Sheikh, 2021). They are also computer hardware and software experts who break into the security of stealing or damaging the important or secret information, compromising the security of big organizations, shutting down or altering functions of websites and networks (Levy, 1984). They violate the computer security for their personal gain. They are engaged in various illegal activities, such as stealing data, spreading malware, and disrupting systems (Cornwall, 1986). They sabotage the systems to obtain information about their targets, such as bank information, personal details, phone numbers, etc. (Begum et al., 2016). They can sell the security vulnerability reports to criminal organizations on the black market or use it to compromise computer systems. They can make money by selling data and credit card information on the dark web. They can ruin reputation of anyone to take revenge (Hoffman, 2013).

American computer security consultant and author Kevin Mitnick is one of the most well-known black hat hackers and cybercriminals in the world, who hacked more than forty major corporations, and later he becomes a white hat hacker (Greenberg, 2022). American computer hacker, computer criminal and police informer, Albert Gonzalez is the largest credit card thief in history who has theft more than 170 million of credit card numbers (Stone, 2008). Some countries, such as China, Russia, and the USA hire black hat hacker to steal data related to militaries from other countries (Chng et al., 2022).

6.3 Grey Hat Hackers

The term “grey hat hacker” came into use in the late 1990s by the hacker group L0pht, and was derived from the concepts of the combination of white hat and black hat hackers (Hoffman, 2013). The grey hat hackers occupy a middle ground between the ethical intentions of white hats and the malicious activities of black hat hackers. They attack the device without seeking permission and considered to be illegal, but do not malachite like the black hat hackers (Harris et al., 2004). They neither illegally exploit the vulnerabilities and nor tell anybody how to do so. They sometimes act ethically but sometimes not, and sometimes they repair the vulnerability for a small fee (Regalado et al., 2015). They have skills and intent of the white hat but may break into any system or network without permission. They may disclose vulnerabilities they find without permission, which can be illegal. They work both defensively and aggressively (Nagarani, 2015). The LulzSec Group served as grey hat hacker, and exposed several vulnerabilities in high-profile companies and Government organizations. American well-known computer security researcher Charles Alfred Miller served as grey hat hacker at National Security Agency, iPhone, and Uber (Miller, 2009).

6.4 Green Hat Hackers

Green hat hackers are known as neophytes or newbies. They are the youngest and novice in hacking and cyber security. They have little or no knowledge and experience in hacking at all (Sheikh, 2021). They rely on the use of already written scripts, and ask lots of questions and are typically very curious. Their aim is to learn about cyber security and contribute positively to the field of cyber security (Levy, 1984). They can assist in identifying basic common vulnerabilities and bugs in applications, and have skills in input validation flaws or insecure data storage practices (Gregg, 2006). They can provide valuable feedback from a beginner perspective, and can offer insights into how a less experienced individual might attempt to exploit vulnerabilities that can be informative for developers in understanding potential attack vectors. They can create a controlled environment for learning and identifying security gaps (Hanusch, 2021).

6.5 Red Hat Hackers

Red hat hackers are the ultra-white hat hackers who use extreme and brute force techniques and tactics, and aggressively attacks and disables to combat against malicious black hat hackers through destroying data and systems to stop them (Moore, 2005). They are the greatest concern, danger, and threat for the black-hat community. They use aggressive methods, attack and technique to take down them down (Caldwell, 2011). They typically target Linux systems due to their open-source nature that provides easy access to both command-line interfaces and popular hacking tools. They upload viruses and destroy their devices and computers, or retrieve their information and kill off their devices (Levy, 1984). They are vigilantes in cyber security, and usually they hack Government agencies and top-secret headquarters in order to gain sensitive information (Chng et al., 2022).

6.6 Script Kiddie

A script kiddie is a new and unskilled hacker to the world of hacking, and exclusively relies on copying codes and scripts to perform malicious code injection and modification. S/he is an amateur hacker with limited technical knowledge, and relies on already-made programs in order to use them (Hanusch, 2021). S/he simply wants to buy all the tools s/he needs and uses them without doing anything themselves. S/he can use pre-made tools and scripts to launch attacks without understanding the underlying mechanisms. In some cases, script kiddies watch tutorial videos in order to apply them and use them for themselves (Levy, 1984). Sometimes s/he may rather pay money and performs a straightforward attack than come up with new, secret, or creative ways to carry out his/her plans (Banda et al., 2019).

6.7 Blue Hat Hackers

Blue hat hackers are external cyber security professionals hired by an organization to find vulnerabilities in a system before its public launch, who violate laws or ethical standards for nefarious purposes, such as cybercrime, cyber warfare, and malice (Cornwall, 1986). They are as like script kiddies but want to take revenge against someone using hacking as a tool (Banda et al., 2019). But they are unlike green hat hackers and have no desire to learn new techniques and rely on the knowledge that they have already gained to perform their tricks. They are classified as invited security professionals to find any vulnerability in Microsoft Windows (Levy, 1984).

There are many skillful hackers that are present in the different countries but some countries are in the leading position in the field of hacking, such as China, the USA, Turkey, Russia, Taiwan, Brazil, Romania, India, Italy, Hungary, etc. (Cyware News, 2020).

7. Tools of Hacking

Kali Linux offers more than 600 preinstalled tools for ethical hacking. Some of the best use tools are Nmap, Metasploit, Nessus, Hydra, and Wireshark, and these are used by security professionals to find and fix vulnerabilities (Mohajan, 2025b).

7.1 Nmap

Network mapper (Nmap) is a free and open-source network vulnerabilities scanner that is used by penetration testers. It is an integral part of academic activities that provides more advanced service detection, vulnerability detection, and other features. It is a tool that can be used to discover services running on internet connected systems. It is also used by ethical hackers to assess their own networks for vulnerabilities (Haines et al., 2003).

The Nmap is created by American network security expert Gordon Lyon. It detects the vulnerabilities of the network and presents the report to the user (Lyon, 2008). It is an open source Linux command line tool that is used to scan Internet Protocol (IP) addresses and ports in a network and to identify hosts, services, operating systems, and firewall (Joshi, 2021). It is developed to scan big networks, and beneficial for network inventory, controlling service updates schedules, and monitoring host or service uptime. It is compatible with all major operating systems, such as Windows, Mac OS X, and Linux (Elhai & Hall, 2016). It helps the device to quickly map out a network without sophisticated commands, and supports simple commands and complex scripting through the Nmap scripting engine. It performs a basic port scan for fast result that identifies hosts on a network, and interrogates network services on remote devices to determine application name and version number (Medeiros et al., 2009).

7.2 Wireshark

Wireshark is a free and open source network scanner, which is one of the best packet analyzers available today that detects the vulnerabilities of the network. It was developed by Gerald Combs, Director of Open Source Projects at Riverbed Technology, USA; under the name Ethereal in 1997, and later it was renamed Wireshark in May 2006 due to trademark issues (Sanders, 2007). It is available for UNIX and Windows that capture live packet data from a network interface. It is a network packet analyzer that presents captured packet data in as much detail as possible (Lamping, 2004).

The Wireshark scanner can capture traffic from many different network media types, such as Ethernet, Wireless LAN, Bluetooth, Universal Serial Bus (USB), etc. It provides deep visibility into network communications, allowing users to analyze data at the packet level that seeks to simplify and enhance the process of network traffic analysis (Cheok, 2014). It is the world's leading network protocol analyzer that is used by network administrators, security professionals, and developers to capture, inspect, and troubleshoot network traffic in real time (Jain, 2022). It is widely used for network troubleshooting, open source network analysis, software and communications protocol development, and education that can capture and display real-time details of network traffic (Orebaugh et al., 2007).

7.3 Metasploit

The Metasploit tool is a computer security project that provides information about security vulnerabilities and

aids in penetration testing and intrusion detection system (IDS) signature development for creating security tools and exploits (Kennedy et al., 2011). It consists of tools, libraries, modules, and user interfaces, and the basic function of it is a module launcher that allows the user to configure an exploit module and launch it at a target system (Ajero, 2022). It allows the operator to easily build attack vectors to augment its exploits, payloads, encoders, and more in order to create and execute more advanced attacks. It is not just a tool but an entire framework that provides the infrastructure needed to automate mundane, routine, and complex tasks (Muñoz et al., 2018). It is available in two versions: commercial and free that are used with either a command prompt or a web interface (Foster & Liu, 2006).

The Metasploit is created by American network security expert, open source programmer and hacker H. D. Moore in 2003 as a portable network tool using Perl. With the help of Spoonm, he has released a total rewrite of the project, Metasploit 2.0, in April 2004 that includes 19 exploits with 27 payloads. In 2009, it is owned by a US-based cyber security company Rapid7, a leader in the vulnerability-scanning field. It operates as an open-source project and accepts contributions from the community through GitHub.com pull requests (Foster & Price, 2005). At present it has more than 2,074 exploits, organized under the platforms on AIX, Android, iPhone, BSD, BSDi, Cisco, Firefox, FreeBSD, HP-UX, Irix, Java, JavaScript, Linux, mainframe, NetBSD, NetWare, NodeJS, OpenBSD, macOS, Maemo, PHP, Python, R, Ruby, Solaris, Unix, and Windows (Rahalkar & Jaswal, 2017). It is widely used worldwide in network security professionals, system administrators, product vendors, and security researchers to perform penetration tests, verify patch installations, and perform regression testing (Teixeia et al., 2021).

7.4 Nessus

Nessus is a platform developed by Tenable Holdings, Inc., a cyber-security company based in Columbia, Maryland that scans for security vulnerabilities in devices, applications, operating systems, cloud services, and other network resources (Medhora & Finkle, 2016). In 1998, Renaud Deraison, the father of the Nessus vulnerability scanner, has created *The Nessus Project* as a free remote security scanner. It features high-speed asset discovery, configuration auditing, target profiling, malware detection, sensitive data discovery, etc. (LeMay, 2005).

It identifies software flaws, missing patches, malware, and misconfiguration errors across a wide range of operating systems, devices, and applications used in organizations (Mohajan, 2025e). It supports more technologies than competitive solutions, scanning operating systems, network devices, next generation firewalls, hypervisors, databases, web servers, and critical infrastructure, such as Common Vulnerability Scoring System (CVSS) v4, Exploit Prediction Scoring System (EPSS), and Tenable's Vulnerability Priority Rating (VPR) feature; for vulnerabilities, threats, and compliance violations (Awati, 2025). It is now available in two enterprise versions: Nessus Professional and Nessus Expert that support unlimited IT vulnerability assessments and multiple systems for vulnerability scoring. It provides more than 450 preconfigured templates to help users understand where vulnerabilities are present (Tenable, 2025).

7.5 The Hydra

Hydra (THC-hydra) is a powerful and versatile password cracking tool for testing the security of authentication systems in social media, such as Facebook, Instagram, twitter, etc. (Kumar & Agarwal, 2018). It is very fast, stable, and flexible network login hacking tool. It is an open-source tool designed to perform cyber-attacks through brute-force on various protocols and services to test the authentication mechanisms (Groza, 2009). It is easily available online at GitHub where all its newest releases are frequently updated. It is developed by German IT security expert and ethical hacker Van Hauser (Marc Heuse) (Yiannis, 2013). Some top uses of Hydra are password cracking, brute force attacks, dictionary attacks, secure shell (SSH) login testing, file transfer protocol (FTP) login testing, simple mail transfer protocol (SMTP) authentication testing, database login testing, web application login testing, voice over IP (VoIP) authentication testing, network device testing, etc. (Mamber, 1996).

The Hydra is a parallelized network login cracker built into various operating systems, such as Kali Linux, Parrot and other major penetration testing environments. It is capable of rapidly guessing and applying numerous password combinations to uncover authentication credentials across a variety of protocols (Dhanjani & Clarke, 2005). It is commonly used by penetration testers with a set of programs, such as crunch, cup, etc. Although it is widely used by cyber security professionals, its use must always be governed by legal authorizations to prevent abuse. It remains an essential tool in the arsenal of security testing (McNab, 2011). It supports more than fifty common login protocols on websites, such as FTP, SMB, POP3, IMAP, MySQL, VNC, SSH, HTTP(S), Cisco AAA, Cisco auth, Cisco enable, CVS, HTTP-Proxy, ICQ, IRC, LDAP, MS-SQL, NNTP, Oracle Listener, Oracle SID, PC-Anywhere, PC-NFS, PostgreSQL, RDP, Rexec, Rlogin, Rsh, SIP, SMTP, SMTP Enum, SNMP v1+v2+v3, SOCKS5, Subversion, Telnet, VMware-Auth, and XMPP (Kakarla et al., 2018).

8. Types of Hacking

Hacking is the activity of characterizing weaknesses in a knowledge processing system and a network to take advantage of the security to comprehend access to private knowledge through the exploitation of computers to commit fallacious acts like fraud, privacy invasion, stealing corporate/personal knowledge, etc. (Pal, 2016). Some devices that are at risk from hackers are smartphones, IoT gadgets, and older computers (Mohajan, 2015f). The computer and network hacks come in many forms and some of the most prevalent types are backdoors, denial of service (DoS), distributed DoS, phishing attacks, etc. (Gada, 2024).

8.1 Backdoors

When hackers can make an unapproved approach to any network by using a sound easy route without even being noticed by any security software, then it is known as backdoor hacking that contains the characteristics of all strikes (Li et al., 2022). A backdoor attack is a typically covert method of bypassing normal authentication or encryption in a computer, product, embedded device, or its embodiment, such as cryptosystem, algorithm, chipset, etc. (Eckersley & Portnoy, 2017). It represents a threat actor where the attackers can take over system resources, go through networks, and set up various malware programs. It is used for securing remote access to a computer, or obtaining access to plaintext in cryptosystems. A developer may create a backdoor so that an application, operating system or data can be accessed for troubleshooting (Ashok, 2017).

It may take the form of a hidden part of a program, a separate program, code in the firmware of the hardware, and parts of an operating system, such as Windows (Li et al., 2021). It may lead to data breaches, financial losses, reputational damage, and concerns about national security. Various types of malware are used in backdoor attacks, such as Cryptojacking, DoS attacks, Ransomware, Spyware, Trojan horse, federated learning, hardware, internet of things (IoT), Island hopping, Phishing, Steganography, etc. (Krieg et al., 2013).

8.2 Denial of Service

Denial of Service (DoS) attack is one of the major threats and is considered as one of the hardest problem in the internet today. It makes hacker capable of breaking any network without getting access in the network that targets the email or transmission control protocol (Elleithy et al., 2005). It is any type of attack on a networking structure to disable a server from servicing its clients that range from sending millions of requests to a server in an attempt to slow it down, flooding a server with large packets of invalid data, and to sending requests with an invalid (Fulp et al., 2001).

There are different types of DoS attacks, such as i) flood attack, that is flooding the target machine with external communications requests, so that it cannot respond to legitimate traffic, or responds so slowly as to be made unavailable; ii) logic and software attacks, that are internet packets are sent that should use bugs in the software (Nagesh & Sekaran, 2006); iii) Ping of Death, which is simulated against a Microsoft Windows 95 computer, iv) TCP SYN Flood, which is simulated against a Microsoft Windows 2000 IIS FTP Server, and v) distributed DoS (Mohajan, 2025e).

The DoS is an attack in which one or more machines target a victim that attempts to partially or completely prevent the victim from doing useful work, and to stop from viewing portions of the internet (Prakash et al., 2016). It clogs up so much memory on the target system that it cannot serve its users, or it causes the target system to crash, reboot, or otherwise deny services to legitimate users. As a result, a legitimate user or organization is deprived of certain services, such as web, email, or network connectivity that the user would normally expect to have. It poses significant threats to network security, disrupting critical services by overwhelming targeted systems with malicious traffic (Jain & Singh, 2012). It cannot be stopped or prevented, but some precautionary measures are taken into consideration to make the attacker very hard to attack. The network architecture should be built in a stronger way to secure the resources against various attacks. The host computers must be updated with the latest security patches and techniques (Bhardwaj et al., 2016).

8.3 Distributed DoS (DDoS)

The DDoS is demonstrated by simulating a distribution zombie program that will carry the Ping of Death attack. It makes the DoS strike efficient with number of sources and be controlled remotely. It has the characteristics of network foundation strike and operating system strike (Gresty et al., 2001). It is either flood attack or logic attack, but it uses many people, different computers, often from thousands of hosts infected with malware, and bots under the attacker's control, and usually attacks target sites, such as banks, credit card payment gateways, etc. (Dzaferovic et al., 2020). It occurs when multiple systems flood the bandwidth or resources of a targeted system, usually one or more web servers. It is analogous to a group of people crowding the entry door of a shop, making it hard for legitimate customers to enter, thus disrupting trade and losing the business money (Prince, 2016).

The first DDoS attack shutdown the entire internet access on the city for a couple of hours happened in 1997

during a hacker's conference event in Las Vegas by the well-known spammer and hacker Khan C. Smith. Revenge and blackmail as well as hacktivism can motivate these attacks (Lohachab & Karambir, 2018). Some common examples of DDoS attacks are UDP flooding, SYN flooding, and DNS amplification. The most obvious symptom of a DDoS attack is a site or service suddenly becoming slow or unavailable (Bhattacharyya & Kalita, 2016). If the attacks are from multiple sources, it can be difficult for the host to identify and stop those. One solution available to virtually all network admins is to create a blackhole route and funnel traffic into that route (Zuckerman et al., 2011).

8.4 Phishing Attacks

The most talked about topic in the world of social engineering is phishing that is the most common type of cybercrime. Phishing is a luring type cyber-attack that thieves use to "fish for" unsuspecting internet users' personal identifying information through emails and mirror-websites that the messages appear to come from well-known and trustworthy websites, and pressure has to act quickly, without thinking (Wright, 2016). It is a type of cybercrime where malicious actors trick victims into revealing sensitive information, such as usernames, passwords, credit card details, and personal data, by posing as a trustworthy entity (Ramzan, 2010). Attackers use deceptive emails, text messages, and fake websites to lure people into clicking malicious links or downloading malware, such as viruses, worms, adware, or ransomware (Jansson & von Solms, 2011).

The term "phishing" was coined by well-known spammer and hacker Khan C. Smith using the hacking tool AOHell that was released in 1994. Phishing may be various types, such as email phishing, spear phishing, whaling phishing, page hijacking, voice phishing, QR code phishing, SMS phishing, man-in-the-middle phishing, etc. Spear phishing is an email-spoofing attack that targets a specific organization or an individual, seeking unauthorized access to sensitive information. On the other hand, the whaling phishing is a type of fraud that targets high-profile end users, such as C-level corporate executives, politicians, and celebrities (Lin et al., 2019). The victims are asked to disclose name, parents name, place of birth, credit card numbers, social security numbers, account numbers, passwords, and other private information (Jansson & von Solms, 2011). Sometimes fake emails come from a reputable and recognize company that offers business proposal. The provided link appears to be the official website of the company, although it is fraudulent (Olivo et al., 2011). Anarchist hacking is performed by script kiddie that is not a professional type of hacking, because it cannot deal with any unapproved access. It makes strike only on any web content, such as Facebook, Instagram, and twitter (Son & Kim, 2008).

8.5 Password Cracking

Passwords are a system designed to provide authentication (Yan et al., 2004; Mohajan, 2025g). Human-memorable passwords remain a common form of access control to data and computational resources (Narayanan & Shmatikov, 2005). Password cracking is an attack vector that involves hackers attempting to determine a password for unauthorized authentication (Hellman, 1980). This can be done in several ways: i) brute-force attacks, which involves trying all possible combinations of characters until the correct password is found, it is time-consuming and requires significant computational power (Bahadursingh, 2020); ii) dictionary attacks, which involves the use of a dictionary of common passwords, and the hacker systematically tries each entry in the hope that the user has used a common password (Lundin, 2013); and iii) rainbow tables, which involves pre-computing the hashes for possible passwords and storing them in a 'rainbow table' (Burr et al., 2006). Password cracking can also do through other tactics, such as by memory-scraping malware, shoulder surfing, third party breaches, and tools like Redline password stealer (Oechslin, 2003).

8.6 Keyloggers

A keylogger is a type of spyware that records keystrokes of a user on the keyboard, and saves them to a log file; and the hackers use them to steal sensitive information, such as usernames, passwords, credit card numbers, messages, and other personal confidential information (Nyang et al., 2014). It is a smart malicious binary that takes an action to record keys that the user presses when the system is active at sensitive times. It can be used to study keystroke dynamics or human-computer interaction, and extremely useful for keeping track on ongoing criminal activity. It can be installed through malicious downloads, such as email attachments or by physically connecting a hardware device to a computer (Leyden, 2000).

There are two main types of keyloggers: hardware keyloggers and software keyloggers. A hardware keylogger is a physical device that is connected directly to the keyboard and the computer through manually using one of two approaches (Wajahat et al., 2019). It has a non-volatile memory device, such as flash memory that stores the recorded data, and retaining it even when power is lost; and a microcontroller that interprets the datastream between the keyboard and computer, processes it, and passes it to the non-volatile memory (Kuncoro & Kusuma, 2018). It may be different types, such as i) regular hardware keylogger, which is attached between the computer keyboard and the computer; ii) wireless keylogger sniffers, which collect packets of data being transferred from

a wireless keyboard and its receiver and then attempt to crack the encryption key being used to secure wireless communications between the two devices (Arghire, 2019); iii) firmware, which is responsible for handling keyboard events, and can be reprogrammed so that it records keystrokes as it processes them; iv) keyboard overlays, which is a fake keypad is placed over the real one so that any keys pressed are registered by both the eavesdropping device as well as the legitimate one that the customer is using (Wajahat et al., 2019); and v) key commands, which programs require keyloggers to know when the user is using a specific command. In the mid-1970s, the Soviet Union developed and deployed a hardware keylogger targeting US Embassy typewriters (Kirk, 2008).

A software-based keylogger is a computer program that can be installed without having direct access to the computer. It captures data when it travels across the keyboard and through the operating system that keeps track of keystrokes, saves them in a secure location, and subsequently sends them to the keylogger author (Aslam et al., 2004). Interrogation cycle, traps keylogger, rootkits keylogger, and keylogger kernel mood are the four primary categories of software keylogger. An early keylogger was written by Perry Kivolowitz and posted to the Usenet newsgroup net.unix-wizards, net.sources on November 17, 1983 (Le et al., 2008).

The main purpose of keyloggers is to tamper with the chain of events that occur when a key is pressed, and information is displayed on the screen as a result of the keystroke that can be used for both lawful and illegitimate objectives, depending on the user who is utilizing it (Berninger, 2012). Keylogger is common with many Trojans that is designed to mimic legitimate software and bypass anti-virus or anti-malware scanners (Javaheri et al., 2018). It is a sophisticated tool that can access not only the platform, but also the user's private information like their name, password, PIN, and card and bank statement. The aim of a keylogger is to act as a mediator between the system functions and user requests they come to know through discovering the pressed keys (Le et al., 2008).

8.7 Buffer Overflows

In programming and information security, buffer overflow is overloading a buffer within the memory of a system with more data than it is designed to handle. It is a software coding error that can be exploited by hackers to gain unauthorized access to corporate systems (Gerg, 2005). It occurs when data written to a buffer corrupts data values in memory addresses adjacent to the destination buffer due to insufficient bounds checking (Gupta, 2012). It also takes place when an attacker manipulates the coding error to carry out malicious actions and compromise the affected system (Lhee & Chapin, 2003). This can cause the system to crash or allow a hacker to execute arbitrary code. Various buffer overflow techniques have been discovered and numerous incidents of buffer overflow attacks have been happened. The techniques to exploit buffer overflow vulnerability vary by architecture, operating system, and memory region (Cowan et al., 1998). There are five main types of buffer overflows as,

i) Stack-based overflows: These are the most common and involve overloading the stack. These occur when an attacker sends data containing malicious code to an application, which store the data in a stack buffer (ALHusayn & Alsuwat, 2020).

ii) Heap-based overflows: These are more difficult to carry out than the stack-based approach. These target the heap, and attack flooding a program's memory space beyond the memory it uses for current runtime operations (Cowan et al., 2000).

iii) Format string attack: It takes place when an application processes input data as a command or does not validate input data effectively. It enables the attacker to execute code, read data in the stack, or cause segmentation faults in the application (Akritidis et al., 2005).

iv) Integer buffer overflow: It is the mathematical process of an integer overflow leads to an integer which is bigger for the integer that can hold. It is divided into three categories based on undersigned structural, symbolic and truncation issues underflow and overflow (Ren et al., 2019).

v) Unicode buffer overflow: It appears by adding Unicode characters into an entry expecting ASCII characters. When ASCII code just includes Western language characters, Unicode can initial a character for all languages (Shahab et al., 2020).

8.8 Privilege Escalation

Privilege escalation is a hacker gaining higher levels of access to a system than originally intended, often with the goal of gaining full control by exploiting vulnerabilities, misconfigurations, or weak controls (Kuzuno & Yamauchi, 2024). It occurs when the user is able to obtain a higher level of access than an administrator intended, possibly by performing kernel level operations (Yamauchi et al., 2021). There are two main types of privilege escalation: i) vertical privilege escalation, where a hacker starts with a low-level account and exploits a vulnerability to gain a higher-level account, such as an administrator account; and ii) horizontal privilege

escalation, where a hacker uses his existing account level to access resources that should be off-limits (Mehmood et al., 2023).

Vertical privilege escalation occurs when the user is able to obtain a higher level of access than an administrator intended, possibly by performing kernel level operations (Kim & Lee, 2023). Horizontal privilege escalation occurs when an application allows the attacker to gain access to resources which normally would have been protected from an application (Kuzuno & Yamauchi, 2024). It needs no upgrading the privilege of accounts, and often relies on the bugs in the system. The result is that the application performs actions with the same user but different security context (Diogenes, 2019).

9. Steps of Hacking

In information security, hacking refers to exploiting vulnerabilities in a system and compromising its security to gain unauthorized access and control. In ethical hacking the goal of this simulated attack is to uncover the weak points of the organizations and suggest ways to strengthen these (Makwana et al., 2025). It encompasses a range of activities, from simple password guessing to sophisticated attacks exploiting vulnerabilities in networking. There are mainly five phases in hacking and a hacker no need to follow these five steps in a sequential manner.

9.1 Reconnaissance

Reconnaissance is the preparation stage and is also called as Footprinting and information gathering phase, where the hackers try to gather as much information as possible on the targeted computer system and network (Bendjakhdel, 2019). This involved researching the target infrastructure, identifying potential vulnerabilities, and understanding the defenses of the system. The primary objective of it is to understand the target environment, infrastructure, and potential weak points, such as finding out the internet protocol (IP) address range, network, domain name system (DNS) records, etc. of the targets (Sanghvi & Dahiya, 2013).

Usually, the information is collected in three groups: network, host, and people involved. There are two types of Footprinting: i) active which directly interacts with the target to gather information about the target, ii) passive which tries to collect the information about the target without directly accessing the target, such as from social media, public websites, etc. (Shah et al., 2019).

9.2 Scanning

Scanning is the pre-attack stage that is done on the basis of information gathered during the reconnaissance phase to identify Internet Protocol (IP) addresses, open ports, live systems, running services, live hosts, and vulnerable services (Bou-Harb et al., 2014). It begins by inputting the root Uniform Resource Locator (URL), performing authentication if required, and then navigating from page to page to identify other URLs and elements, such as forms. But the system must provide a mechanism to identify already visited links to prevent the process from continuing indefinitely (Odion et al., 2023). It is a structured and deliberate process that aims to find the potential weaknesses in one organization's environment immersed in internet security. It works like a security health check for digital systems (Pandey & Chaudhary, 2022).

It helps ethical hackers to map the network, detect live machines, understand topology, identify weak points, and planning simulated attacks for testing defenses. It tries to increase the organization's defenses, reducing the likelihood of successful cyber-attacks and minimizing the impact of any potential security incidents (Bou-Harb et al., 2014). There are three types of scanning:

i) Port scanning: It involves scanning the target for the information, such as open ports, live systems, and various services running on the host. It is used to scan a network for open ports that can be potential entry points for attackers, such as Nmap (RSI Security, 2023).

ii) Network vulnerability scanning: It discovers open ports and identifies any unfamiliar services that are using them. It checks the target for known vulnerabilities and misconfigurations that can be exploited, for example, Nessus (RiskOptics, 2022).

iii) Web application scanner: It finds the topology of network, routers, firewalls servers if any, and host information and drawing a network diagram with the available information that serves as a valuable piece of information throughout the hacking process, such as SQL injection, cross-site scripting, etc. (Mohaidat & Al-Helali, 2024).

iv) Database scan: It assess the security of database systems by scrutinizing setup, access controls, and stored data for vulnerabilities, such as insecure permissions, injection issues, and unsafe configurations. It offers valuable insights to enhance database security and protect sensitive information (Pandey & Chaudhary, 2022).

v) Source code scan: It is crucial to proactively assess source code for security vulnerabilities to address issues before they become costly to correction that analyzes software applications' source code, detecting security flaws, coding errors, and vulnerabilities. It focuses on discover the potential issues like input validation errors,

improper coding practices, and the use of vulnerable libraries (Grance et al., 2003).

vi) Cloud vulnerability scan: It assesses the security of cloud environments, such as IaaS, PaaS, and SaaS installations, providing recommendations to enhance deployment security. It examines cloud configurations, access restrictions, and services to discover misconfigurations, inadequate security practices, and vulnerabilities specific to cloud platforms (Basan, 2023).

9.3 Gaining Access

Gaining access is the third stage where the attacker gets access to the computer system. He uses malware to enter the point-of-sale (POS) systems and harvests card details (Stallings & Brown, 2018). The techniques used during this stage can vary widely, depending on the specific vulnerability being exploited (Vest & Tubberville, 2019). Once a potential entry point has been identified, the hacker takes attempt to exploit the identified vulnerabilities to gain unauthorized access to the system (Islam et al., 2023).

9.4 Maintaining Access

Maintaining access is the process of sustaining accessibility to the computer system once access has been gained. In this case, malware is installed to continuously capture payment data (Griffiths, 2022). This can be done using Trojans, rootkits, backdoors or other malicious files until he finishes the tasks he planned to accomplish in that target even after system restarts, password changes, or other defensive measures (Martin, 2019). This stage is crucial for a hacker, as it allows him to continue exploiting the system even if his initial entry point is closed (Kaur et al., 2023).

9.5 Cleaning Tracks

Clearing tracks are hiding one's malicious acts to prevent being uncovered. An intelligent hacker always clears all evidence so that in the later point of time and no one will find any traces leading to him (Martin, 2019). This can be done through the deleting the log files that record intrusion events; altering modifying, and corrupting registry values; uninstalling all applications used during exploitation; deleting all folders created during the attack; and removing any trace of the attack (Ahn et al., 2010). If a hacker has successfully gained access to the networking system, exploited vulnerabilities, and exited without detection. For example, in 2019 in the Capital One breach, the hackers tried to hide their AWS activity logs but were eventually tracked through cloud service records (Clancy, 2022).

10. Conclusions

This paper deals with the basic ideas of unethically hacking that is used the skills to harm others. At present hacking is a big problem faced by Governments, companies, and private citizens around the world. Cybercriminal hackers are stealing billions of dollars, and cyber security has endangered worldwide. Hacking is an elite profession within the IT field that requires an extensive and detailed understanding of IT concepts and technologies. The internet allows the hackers to take files, programs, passwords, and other information from users, and sometimes users cannot understand it. To protect a networking system from hacker attacks, an organization needs to know the way of hacker's thinking and methodology, as well as about the tools, which s/he can use.

References

Ahn, G., et al. (2010). Representing and Reasoning about Web Access Control Policies. Proceedings of the 34th Annual IEEE International Computer Software and Applications Conference, COMPSAC 2010, Seoul, Korea, 19-23 July 2010.

Ajero, G. L. (2022). Differentiate Metasploit Framework Attacks from Others. M. Sc. Thesis, Stephen F. Austin State University.

Akritidis, P., et al. (2005). STRIDE: Polymorphic Sled Detection through Instruction Sequence Analysis. Proceedings of the 20th IFIP International Information Security Conference (IFIP/SEC 2005). IFIP International Information Security Conference.

ALHusayn, S. M. S., & Alsuwat, E. (2020). The Buffer Overflow Attack and How to Solve Buffer Overflow in Recent Research. *Academic Journal of Research and Scientific Publishing*, *2*(19), 1-13.

Arghire, I. (2019). Business Users Targeted by HawkEye Keylogger Malware. *Security Week*, 28 May 2019.

Ashok, I. (2017). Hackers using NSA malware DoublePulsar to infect Windows PCs with Monero mining Trojan. *International Business Times UK*.

Ashwini, S. K., & Thippeswamy, K. (2018). A Brief Information of Ethical Hacking. 3rd National Conference on Image Processing, Computing, Communication, Networking and Data Analytics.

Aslam, M., et al. (2004). Antihook Shield against the Software Keyloggers. In Proceedings of the National

Conference of Emerging Technologies.

Awati, R. (2025). What is the Nessus Vulnerability Scanning Platform? TechTarget and Search Networking.

Baglione, L. (2012). *Writing a Research Paper in Political Science*. Thousand Oaks, California: CQ Press.

Bahadursingh, R. (2020). A Distributed Algorithm for Brute Force Password Cracking on n Processors. doi:10.5281/zenodo.3612276

Banda, R., et al. (2019). Technological Paradox of Hackers Begetting Hackers: A Case of Ethical and Unethical Hackers and their Subtle Tools. *Zambia ICT Journal*, *3*(1), 40-51.

Basan, M. (2023). 12 Types of Vulnerability Scans & When to Run Each. eSecurityPlanet.

Begum, S., et al. (2016). A Comprehensive Study on Ethical Hacking. *International Journal of Engineering Sciences & Research Technology*, *5*(8), 214-219.

Bendjakhdel, S. E. (2019). Reconnaissance Study: Concept and Design. *El-Khaldounia Journal of Human and Social Sciences*, *11*(1), 10-17.

Berninger, V. W. (2012). *Past, Present, and Future Contributions of Cognitive Writing Research to Cognitive Psychology*. New York/Sussex: Taylor & Francis.

Bhardwaj, A., et al. (2016). *Three Tier Network Architecture to Mitigate DDOS Attacks on Hybrid Cloud Environments*. ACM Computing surveys.

Bhardwaj, M., & Singh, G. P. (2011). Types of Hacking Attack and their Counter Measure. *International Journal of Educational Planning & Administration*, *1*(1), 43-53.

Bhattacharyya, D. K., & Kalita, J. K. (2016). *DDoS Attacks: Evolution, Detection, Prevention, Reaction, and Tolerance*. Boca Raton, FL: CRC Press.

Bou-Harb, E., et al. (2014). Cyber Scanning: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials*, *16*(3), 1496-1519.

Burr, W. E., et al. (2006). Electronic Authentication Guideline. Gaithersburg, MD: National Institute of Standards and Technology.

Caldwell, T. (2011). Ethical Hackers: Putting on the White Hat. *Network Security*, *2011*(7), 10-13.

Cekerevac, Z., et al. (2018). Hacking, Protection and the Consequences of Hacking. *Communications*, *20*(2), 68-72.

Cheok, R. (2014). *Wireshark: A Guide to Color My Packets*. SANS Institute.

Chirillo, J. (2002). *Hack Attacks Revealed: A Complete Reference with Custom Security Hacking Toolkit*. John Wiley & Sons.

Chng, S., et al. (2022). Hacker Types, Motivations and Strategies: A Comprehensive Framework. *Computers in Human Behavior Reports*, *5*(2022), 100167.

Clancy, R. (2022). What is Broken Access Control Vulnerability? EC-Council.

Cornwall, H. (1986). *Hacker's Handbook*. Publisher: E Arthur Brown.

Cowan, C., et al. (1998). StackGuard: Automatic Adaptive Detection and Prevention of Buffer-Overflow Attacks. Proceedings of the 7th USENIX Security Symposium USENIX: San Antonio, TX, January 1998, pp. 63-77.

Cowan, C., et al. (2000). Buffer Overflows: Attacks and Defenses for the Vulnerability of the Decade. Proceedings DARPA Information Survivability Conference and Exposition Hilton Head, SC, January 2000; 119-129.

Creswell, J. W. (2013). *Review of the Literature*. *Research Design. Qualitative, Quantitative, and Mixed Method Approaches* (4th Ed.). Thousand Oaks, California: SAGE Publications.

Cyware News. (2020). *Top 10 Countries with Most Hackers in the World*. Cyware Social.

Dhanjani, N., & Clarke, J. (2005). *Network Security Tools: Writing, Hacking and Modifying Security Tools*. Publisher: O'Reilly Media, Inc.

Diogenes, Y. (2019). *Cybersecurity: Attack and Defense Strategies.* Erdal Ozkaya, Safari Books Online (2nd Ed.), Packt.

Dzaferovic, E., et al. (2020). DoS and DDoS vulnerability of IoT: A review. *Journal of Sustainable Engineering and Innovation*, *1*(1), 43-48.

Eckersley, P., & Portnoy, E. (2017). Intel's Management Engine is a Security Hazard, and Users Need a Way to

Disable It. Electronic Forum Foundation.

Elhai, J. D., & Hall, B. J. (2016). Anxiety about Internet Hacking: Results from a Community Sample. *Computers in Human Behavior*, 54, 180-185.

Elleithy, K., et al. (2005). Denial of Service Attack Techniques: Analysis, Implementation and Comparison. *Journal of Systemics, Cybernetics and Informatics*, *3*(1), 66-71.

Foster, J. C., & Liu, V. (2006). *Writing Security Tools and Exploits*. Syngress Publishing, Inc., Canada.

Foster, J. C., & Price, M. (2005). *Sockets, Shellcode, Porting, and Coding: Reverse Engineering Exploits and Tool Coding for Security Professionals*. Syngress Publishing, Elsevier.

Fuchs, C. (2014). Anonymous: Hacktivism and Contemporary Politics. In: Idem (Ed.), *Social Media, Politics and the State*, pp. 88-106. New York: Routledge.

Fulp, E., et al. (2001). Preventing Denial of Service Attacks on Quality of Service. DARPA Information Survivability Conference and Exposition (DISCEX II'01), II(2), June 2001, pp. 159-172.

Gada, P. D. (2024). The Art of Hacking. *Journal of Emerging Trends and Novel Research*, *2*(12), a73-a84.

Galvan, J. L. (2015). *Writing Literature Reviews: A Guide for Students of the Social and Behavioral Sciences* (6th Ed.). Pyrczak Publishing.

Gerg, I. (2005). An Overview and Example of the Buffer-Overflow Exploit. IAnewsletter. *Information Assurance Technology Analysis Center*, *7*(4), 1-23.

Grance, T., et al. (2003). Guide to Selecting Information Technology Security Products. National Institute of Standards and Technology, NIST Special Publication 800-36.

Greenberg, A. (2022). Kevin Mitnick, Once the World's Most Wanted Hacker, is Now Selling Zero-Day Exploits. Wired.

Gregg, M. (2006). *Certified Ethical Hacker (CEH) Version 9 Cert Guide*. Pearson Publishing. Pearson Education, Inc.

Gresty, D. W., et al. (2001). Requirements for a General Framework for Response to Distributed Denial-ofService. 17th Annual Computer Security Applications Conference (ACSAC'01), December 2001, pp. 422.

Griffiths, C. (2022). The Latest 2022 Cyber Crime Statistics. AAG IT.

Groh, A. (2018). *Research Methods in Indigenous Contexts*. New York: Springer.

Grover, V. K. (2015). Research Approach: An Overview. *Golden Research Thoughts*, *4*(8), 1-7.

Groza, B. (2009). Analysis of a Password Strengthening Technique and Its Practical Use. Third International Conference on Emerging Security Information, Systems and Technologies, Athens, Glyfada, 2009.

Gupta, S. (2012). Buffer Overflow Attack. *IOSR Journal of Computer Engineering*, *1*(2012), 10-23.

Haines, J., et al. (2003). Validation of Sensor Alert Correlators. *IEEE Security & Privacy*, *99*(1), 46-56.

Hanusch, Y. F. (2021). Financial Institutions Should Decline Hackers' Requests for Voluntary Compensation. *South African Journal of Philosophy*, *40*(2), 162-170.

Harris, S., et al. (2004). *Gray Hat Hacking: The Ethical Hacker's Handbook*. McGraw-Hill Osborne Media.

Hellman, M. E. (1980). A Cryptanalytic Time-Memory Trade-Off. *IEEE Transactions on Information Theory*, *26*(4), 401-406.

Hoffman, C. (2013). Hacker Hat Colors Explained: Black Hats, White Hats, and Gray Hats. How-To Geek.

Islam, M. S., et al. (2023). *'Cyber Security'- A Concern for Improving Public Service Delivery: Challenges and Way Forward*. Cabinet Division, Government of the People's Republic of Bangladesh.

Jain, A., & Singh, A. K. (2012). Distributed Denial of Service (DDoS) Attacks: Classification and Implications. *Journal of Information and Operations Management*, *3*(1), 136-140.

Jain, V. (2022). *Wireshark Fundamentals: A Network Engineer's Handbook to Analyzing Network Traffic*. Springer Science and Business Media, New York.

Jansson, K., & von Solms, R. (2011). Phishing for Phishing Awareness. *Behaviour & Information Technology*, *32*(6), 584-593.

Javaheri, D., et al. (2018). Detection and Elimination of Spyware and Ransomware by Intercepting Kernel-level System Routines. *IEEE Access*, 6, 78321-78332.

John, S. (2017). *Scientific Method*. New York, NY: Routledge.

Joshi, S. (2021). What is Nmap and Why You Should Use It? The Hack Report.

Kakarla, T., et al. (2018). A Real-world Password Cracking Demonstration Using Open Source Tools for Instructional Use. *IEEE International Conference on Electro/Information Technology (EIT)*.

Kara, H. (2012). *Research and Evaluation for Busy Practitioners: A Time-Saving Guide*. Bristol: The Policy Press.

Kartalopoulos, S. V. (2008). Differentiating Data Security and Network Security. International Conference on *Communications*, ICC'08, IEEE, pp.1469-1473, 19-23 May 2008.

Kaur, P., et al. (2023). Access Control Application Prevention and Mitigation of Cyber Attacks. *International Journal of Research and Innovation in Applied Science*, *8*(10), 91-105.

Kennedy, D., et al. (2011). *Metasploit: The Penetration Tester's Guide*. San Francisco: No Starch Press.

Kim, Y. M., & Lee, B. (2023). Extending a Hand to Attackers: Browser Privilege Escalation Attacks via Extensions. Proceedings of the 32nd USENIX Conference on Security Symposium, Article No.: 395, pp.7055-7071.

Kirk, J. (2008). Tampered Credit Card Terminals. IDG News Service.

Krieg, C., et al. (2013). Hardware Malware. *Synthesis Lectures on Information Security Privacy and Trust*, *4*(2), 1-115.

Kumar, S., & Agarwal, D. (2018). Hacking Attacks, Methods, Techniques and Their Protection Measures. *International Journal of Advance Research in Computer Science and Management*, *4*(4), 2353-2358.

Kuncoro, P., & Kusuma, B. (2018). Keyloggers is a Hacking Technique That Allows Threatening Information on Mobile Banking User. Third IEEE International Conference on Information Technology, Information System and Electrical Engineering (ICITISEE), Yogyakarta, Indonesia.

Kuzuno, H., & Yamauchi, T. (2024). Mitigation of Privilege Escalation Attack Using Kernel Data Relocation Mechanism. *International Journal of Information Security*, *23*(5), 3351-3367.

Lamping, U. (2004). *Wireshark User's Guide*. For Wireshark 2.1.

Le, D., et al. (2008). Detecting Kernel Level Keyloggers through Dynamic Taint Analysis. College of William & Mary, Department of Computer Science, Williamsburg, VA, Tech.

LeMay, R. (2005). Nessus Security Tool Closes Its Source. CNET.

Levy, S. (1984). *Hackers: Heroes of the Computer Revolution*. Anchor Press/Doubleday Garden City, NY.

Leyden, J. (2000). Mafia Trial to Test FBI Spying Tactics: Keystroke Logging Used to Spy on Mob Suspect Using PGP. The Register.

Lhee, K.-S., & Chapin, S. J. (2003). Buffer Overflow and Format String Overflow Vulnerabilities. *Software: Practice and Experience*, *33*(5), 423-460.

Li, Y., et al. (2021). Backdoor Attack in the Physical World. arXiv:2104.02361v2 [cs.CR] 24 Apr 2021.

Li, Y., et al. (2022). Backdoor Learning: A Survey. arXiv:2007.08745v5 [cs.CR] 16 Feb 2022.

Lin, T., et al. (2019). Susceptibility to Spear-Phishing Emails: Effects of Internet User Demographics and Email Content. *ACM Transactions on Computer-Human Interaction*, *26*(5), 32.

Lohachab, A., & Karambir, B. (2018). Critical Analysis of DDoS: An Emerging Threat over IoT Networks. *Journal of Communication and Information Networks*, *3*(3), 57-78.

Lundin, L. (2013). *PINs and Passwords, Part 2*. SleuthSayers.org. Orlando.

Lyon, G. F. (2008). *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. Insecure.Com LLC (US), California.

Makwana, D., et al. (2025). Ethical Hacking and Hacking Attacks. *Journal of Neonatal Surgery*, *14*(28s), 940-945.

Mamber, U. (1996). A Simple Scheme to Make Passwords Based on One-Way Functions Much Harder to Crack. *Computers & Security*, *15*(2), 171-176.

Martin, J. A. (2019). What is Access Control? A Key Component of Data Security. CSO Online.

McNab, C. (2011). Network Security Assessment: Know Your Network. O'Reilly Media, Inc.

Medeiros, J., et al. (2009). A Data Mining Based Analysis of Nmap Operating System Fingerprint Database.

Computational Intelligence in Security for Information Systems. *Advances in Intelligent and Soft Computing*, 63, 1-8.

Medhora, N., & Finkle, J. (2016). Cybersecurity Software Maker Tenable CEO Steps Down. Reuters.

Mehmood, M., et al. (2023). Privilege Escalation Attack Detection and Mitigation in Cloud Using Machine Learning. *IEEE Access*, 11, 46561-46576.

Memon, I., et al. (2020). The World of Hacking: A Survey. *University of Sindh Journal of Information and Communication Technology*, *4*(1), 31-37.

Miller, C. (2009). *The Mac Hacker's Handbook*. Dai Zovi, Dino. Indianapolis, IN: Wiley.

Mohaidat, A. I., & Al-Helali, A. (2024). Web Vulnerability Scanning Tools: A Comprehensive Overview, Selection Guidance, and Cyber Security Recommendations. *International Journal of Research Studies in Computer Science and Engineering*, *10*(1), 8-15.

Mohajan, H. K. (2017). Two Criteria for Good Measurements in Research: Validity and Reliability. *Annals of Spiru Haret University Economic Series*, *17*(3), 58-82.

Mohajan, H. K. (2018a). Aspects of Mathematical Economics, Social Choice and Game Theory. PhD Dissertation. University of Chittagong, Chittagong, Bangladesh.

Mohajan, H. K. (2018b). Qualitative Research Methodology in Social Sciences and Theoretical Economics. *Journal of Economic Development, Environment and People*, *7*(1), 23-48.

Mohajan, H. K. (2020). Quantitative Research: A Successful Investigation in Natural and Social Sciences. *Journal of Economic Development, Environment and People*, *9*(4), 50-79.

Mohajan, H. K. (2025a). Artificial Intelligence: Prospects and Challenges in Future Progression. *Art and Society*, *4*(7), 38-50.

Mohajan, H. K. (2025b). Machine Learning: A Brief Review for the Beginners. Unpublished Manuscript.

Mohajan, H. K. (2025c). Deep Learning: A Brief Study on Its Architectures and Applications. *Art and Society*, *4*(8), 41-49.

Mohajan, H. K. (2025d). Cybercrime: A Potential Threat to Global Community. Unpublished Manuscript.

Mohajan, H. K. (2025e). Vulnerability of Cyber Security is An Unexpected Threat to Global Internet System. Unpublished Manuscript.

Mohajan, H. K. (2025f). Addiction and Consumption of Cyber Pornography have Increased Risk Factors and Harmful Effects among Emerging Adults. Unpublished Manuscript.

Mohajan, H. K. (2025g). Cyber Child Pornography: An Unexpected Global Heinous Crime against Children. Unpublished Manuscript.

Moore, R. (2005). *Cybercrime: Investigating High Technology Computer Crime*. Matthew Bender & Company.

Muñoz, F. R., et al. (2018). Analyzing the Traffic of Penetration Testing Tools with an IDS. *Journal of Supercomputing*, *74*(11), 6454-646.

Nagarani, C. (2015). Ethical Hacking and Its Value to Security. *Global Journal for Research Analysis*, *4*(10), 163-164.

Nagesh, H. R, & Sekaran, K. C. (2007). Proactive Solutions for Mitigating Denial-of-Service Attacks. *International Journal of Computer Science and Network Security*, *7*(7), 167-175.

Narayanan, A., & Shmatikov, V. (2005). Fast Dictionary Attacks on Passwords Using Time-Space Tradeoff, CCS'05, November 7-11, 2005, Alexandria, Virginia.

Nyang, D., et al. (2014). Keylogging-Resistant Visual Authentication Protocols. *IEEE Transactions on Mobile Computing*, *13*(11), 2566-2579.

Oakley, J. G. (2019). Why Human Hackers? In *Professional Red Teaming*, pp. 15-28, Springer.

Odion, T. O., et al. (2023). VulScan: A Web-Based Vulnerability Multi-Scanner for Web Application. 2023 International Conference on Science, Engineering and Business for Sustainable Development Goals. IEEE Xplore.

Oechslin, P. (2003). Making a Faster Cryptanalytic Time-Memory Trade-Off. Proceedings of Advances in Cryptology (CRYPTO 2003), Lecture Notes in Computer Science, 2729, 617-630, Springer.

Orebaugh, A., et al. (2007). *Wireshark & Ethereal Network Protocol Analyzer Toolkit*. Syngress.

Pal, S. (2016). Overview of Hacking. *IOSR Journal of Computer Engineering*, *18*(4), 90-92.

Pandey, S., & Chaudhary, A. (2022). Vulnerability Scanning. TechRxiv.

Prakash, A., et al. (2016). Detection and Mitigation of Denial of Service Attacks Using Stratified Architecture. *Procedia Computer Science*, *87*(2016), 275-280.

Prince, M. (2016). Empty DDoS Threats: Meet the Armada Collective. CloudFlare.

Rahalkar, S., & Jaswal, N. A. (2017). *Metasploit Revealed: Secrets of the Expert Pentester*. Packt Publishing Ltd., Mumbai.

Ramzan, Z. (2010). Phishing Attacks and Countermeasures. In Stamp, Mark; Stavroulakis, Peter (Eds.). *Handbook of Information and Communication Security*. Springer.

Regalado, D., et al. (2015). *Grey Hat Hacking: The Ethical Hacker's Handbook* (4th Ed.). New York: McGraw-Hill Education.

Ren, J., et al. (2019). A Buffer Overflow Prediction Approach Based on Software Metrics and Machine Learning. *Security and Communication Networks*, *2019*(1), Article ID: 8391425.

RiskOptics. (2022). Vulnerability Scanners: Passive Scanning vs. Active Scanning. https://reciprocity.com/blog/vulnerability-scanners-passive-scanning-vs-active-scanning

RSI Security. (2023). 7 Types of Vulnerability Scanners. RSI Cybersecurity Blog.

Sanders, C. (2007). *Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems*. No Starch Press.

Sanghvi, P. H., & Dahiya, S. M. (2013). Cyber Reconnaissance: An Alarm before Cyber Attack. *International Journal of Computer Applications*, *63*(6), 36-38.

Shah, M., et al. (2019). Penetration Testing Active Reconnaissance Phase: Optimized Port Scanning with Nmap Tool. 2nd International Conference on Computing, Mathematics and Engineering Technologies (ICoMET), 1-6. IEEE.

Shahab, A., et al. (2020). An Automated Approach to Fix Buffer Overflows. *International Journal of Electrical and Computer Engineering*, *10*(4), 3777-3787.

Shanmugapriya, R. (2013). A Study of Network Security Using Penetration Testing. International Conference on Information Communication and Embedded Systems, pp. 371-374, IEEE, 2013.

Sheikh, A. (2021). Introduction to Ethical Hacking. Certified Ethical Hacker (CEH) Preparation Guide. Berkeley, CA: Apress.

Shital, M. M. (2023). *Network and Information Security*. Techknowledge Publication.

Singh, R., & Kumar, S. (2014). Vulnerable Security Aspects of Windows. *International Journal of Engineering Sciences & Research Technology*, *3*(8), 87-92.

Son, J. Y., & Kim, S. S. (2008). Internet Users' Information Privacy-Protective Responses: A Taxonomy and a Nomological Model. *MIS quarterly*, *32*(3), 503-529.

Stallings, W., & Brown, L., (2018). *Computer Security: Principles and Practice* (4th Ed.). United Kingdom: Pearson Education Limited.

Sterling, B. (1992). *The Hacker Crackdown: Law and Disorder on the Electronic Frontier*. Penguin, Bantam Books, New York.

Stone, B. (2008). Global Trail of an Online Crime Ring. *The New York Times*.

Teixeia, D. et al. (2021). *Metasploit Penetration Testing Cookbook*. Packt Publishing.

Tenable (2025). Tenable Nessus 10.7.x User Guide. Tenable Holdings, Inc.

Torraco, R. J. (2016). Writing Integrative Literature Reviews: Using the Past and Present to Explore the Future. *Human Resource Development Review*, *15*(4), 404-428.

Vest, J., & Tubberville, J., (2019). Red Team Development and Operations: A practical Guide. Independently published.

Wajahat, A., et al. (2019). A Novel Approach of Unprivileged Keyloggers Detection. Second IEEE International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 2019.

Wright, A. (2016). The Big Phish: Cyberattacks against US Healthcare Systems. *Journal of General Internal Medicine*, *31*(10), 1115-1118.

Yaacoub, J.-P. A., et al. (2021). Robotics Cyber Security: Vulnerabilities, Attacks, Countermeasures, and Recommendations. *Springer Nature International Journal of Information Security*, *21*(1), 115-158.

Yamauchi, T., et al. (2021). Additional Kernel Observer: Privilege Escalation Attack Prevention Mechanism Focusing on System Call Privilege Changes. *International Journal of Information Security*, *20*(4), 461-473.

Yan, J., et al. (2004). Password Memorability and Security: Empirical Results. *IEEE Security & Privacy*, *2*(5), 25-31.

Yiannis, C. (2013). Modern Password Cracking: A Hands-on Approach to Creating an Optimised and versatile attack. Surrey, Thesis.

Zuckerman, E, et al. (2011). Distributed Denial of Service Attacks Against Independent Media and Human Rights Sites. The Berkman Center for Internet & Society at Harvard University.

Paradigm Academic Press
Innovation in Science and Technology
ISSN 2788-7030
JUN. 2026 VOL.5, NO.2

Design of BIM and IoT-Based Tunnel Construction Monitoring Data Fusion and Safety Early Warning System

Xiaoqing Cheng[1] & Jiangwei Luo[2]

[1] China First Highway Engineering Group Co., Ltd., Beijing, China

[2] China Communications First Highway Engineering Group Xiamen Engineering Co., Ltd., Xiamen, China

Correspondence: Xiaoqing Cheng, China First Highway Engineering Group Co., Ltd., Beijing, China.

doi:10.63593/IST.2788-7030.2026.06.002

Abstract

The tunnel construction environment is complex, and traditional monitoring methods suffer from poor timeliness and insufficient data utilization, making it difficult to support real-time safety management and control during tunnel construction. This paper designs a construction monitoring and early warning system that integrates Building Information Modeling (BIM) and Internet of Things (IoT) technologies, adopting a "cloud-edge-device" three-tier distributed architecture. At the device layer, a multi-source heterogeneous sensor network is deployed; at the edge layer, edge computing gateways are installed for local data preprocessing and real-time analysis; at the platform layer, a data fusion analysis engine and a BIM visualization engine are integrated. For data fusion, the belief Hellinger distance is introduced to improve the Dempster-Shafer (D-S) evidence theory, effectively resolving conflicts among highly contradictory evidence sources. For safety early warning, a "4+4+N" hierarchical index system is established, and a progressively deepened three-level early warning mechanism is constructed, encompassing single-index threshold judgment, multi-parameter fusion evaluation, and Long Short-Term Memory (LSTM) time-series prediction. Finally, the system is validated through field deployment at the Yingeling extra-long tunnel in Hainan Province. The results that the proposed data fusion method achieves an accuracy of 92.5%, a 20.2% improvement over traditional D-S evidence theory. The early warning system attains an accuracy of 92.5% with a false alarm rate of 8.3%, a missed alarm rate of 3.1%, and an average response time of 2.8 seconds providing reliable technical support for tunnel construction safety management.

Keywords: tunnel construction, BIM-IoT collaborative monitoring, multi-source data fusion, improved DS evidence theory, safety early warning

1. Introduction

By the end of 2024, nearly 29,000 highway tunnels have been built and are in operation in China, with a total mileage of over 32,000 km, and the mileage is still growing at a rate of about 2,000 km per year. In 2024, the mileage of extra-long tunnels will account for 31.69%, and the proportion of long and difficult tunnels in new projects is increasing (Ministry of Transport of the People's Republic of China, 2024). As the technical difficulty and scale of tunnel construction continue to grow, disasters such as instability of surrounding rock, collapses, and mud-flooding occur from time to time. Traditional monitoring mainly relies on contact instruments such as levels and convergence meters in combination with manual readings, with limited coverage of measurement points and low monitoring frequency. Data collection and analysis often lag by hours or even longer (Qiao Xiong, Hu Shijing & Tian Zheng, 2025), especially in the construction of extra-long tunnels. As the tunneling mileage increases, the data from a single measurement point cannot reflect the overall state of the surrounding rock, which easily leads to the situation where on-site managers have missed the best time to deal with the danger when they receive the corresponding monitoring data, thereby affecting the construction progress and seriously threatening the lives of personnel.

In recent years, with the continuous development and improvement of BIM (Building Information Modeling) and Internet of Things (iot) technologies, the component semantics, material properties, and spatial relationships carried by BIM provide an intuitive interpretation for the definition of monitoring data, while iot promotes real-time access (Song Xiuguang, Tian Weiyang, Wei Mingzhao, et al., 2026) to sensor networks. It provides new ideas for real-time monitoring of tunnels. Zhao et al. (2026) superimposed BIM with GIS to achieve spatial display and management of railway tunnel monitoring data; Wang Yaqiong et al. (2015) proposed a four-layer architecture for the Internet of Things, using its comprehensive perception, reliable transmission and intelligent assistance capabilities to guide tunnel monitoring and measurement work; Zhang et al. (2019) applied automated real-time monitoring and early warning technology to tunnel engineering and verified its reliability for monitoring the deformation of surrounding rock and support structure; Zhang Jiasong et al. (2021) used laser scanners in conjunction with cloud computing systems in highway tunnel engineering to achieve real-time collection of automated monitoring data and over-limit early warning.

However, there is still a lack of deep data-level integration between BIM and IoT in the above-mentioned research, and a stable association (Wang Chao, Zhou Leisheng, Xu Run, et al., 2020) has not been established between sensor collection values and BIM components; The fusion results of the traditional DS evidence theory are unreliable when the evidence is highly conflicting; Early warning models are often designed in a one-size-fits-all manner, lacking a hierarchical and progressive analysis mechanism (Zhou Yefei, Wang Peng, Xu Linbo, et al., 2026), and thus unable to completely break the problem (He Xuji, 2022) of information silos. Based on the Hainan Yingge Ling Extra-long Tunnel project, this paper focuses on the key issue of how to transform the “massive data” generated into “practical information”, and designs and develops a tunnel construction monitoring data fusion and safety early warning system based on BIM and Internet of Things. The system adopts a three-tier architecture of “cloud - edge - terminal” overall, a multi-source heterogeneous data fusion method based on DS evidence theory was constructed. In terms of early warning, a “4+4+N” index system and a three-layer progressive safety early warning model were built to provide theoretical support and practical reference for the safe advancement of extra-long tunnel construction under complex geological conditions.

2. Overall System Architecture Design

2.1 System Requirements Analysis

Based on the actual situation of domestic under-construction tunnel projects and in light of the actual requirements of the Hainan Yingge Ling extralong Tunnel, it is determined that the system must have the following functions:

(1) Real-time perception of multiple parameters. In addition to arch subsidence and peripheral convergence, parameters such as surrounding rock pressure, support stress, harmful gas concentration, and personnel location should also be taken into account. In the case of the Hainan Yingge Ling Tunnel (8,271 meters in length, double-tunnel separated type), there are an estimated 776 sections of left and right tunnels and shafts, and thousands of monitoring data such as arch settlement and peripheral displacement are expected, with a large amount of data processing.

(2) Data connection. Sensor points, BIM components, geological sections, and construction logs come from different data sources, and the system needs to unify them into the same spatio-temporal framework.

(3) Graded early warning. The early warning system should differentiate severity and provide corresponding handling suggestions. For those with high construction risks, they should be closely monitored to achieve “dynamic and full-process” risk management.

(4) Low power consumption and high reliability. The actual conditions such as poor power supply and communication conditions in the tunnel should be fully considered, and the equipment in the tunnel should have functions such as local caching and retransmission after recovery in case of network interruption.

2.2 System Architecture

For the above requirements, the system architecture adopts a “cloud-edge-end” three-layer distributed system. Specifically, the end layer is equipped with a multi-source heterogeneous sensor network, including laser displacement meters (accuracy ±0.1mm), fiber Bragg grating sensors, multi-gas sensors (CO/CH4/H2S), UWB positioning modules (DWM1000, static accuracy approximately 12.2cm) (Ma Haowei & Xiao Lina, 2025), etc., to achieve comprehensive perception (Liu Xinze, 2024) of the tunnel construction environment. The communication networking uses ZigBee/433MHz for short-range networking and LoRa for long-range backhaul. Considering the unstable signal coverage due to the construction environment in the tunnel, LoRa was chosen as the backbone backhaul on site to ensure stable signal transmission (Wang Yaqiong, Huang Yilin, Wang Kaiyun, et al., 2015) at the maximum burial depth.

Edge computing gateways are deployed in the edge layer to perform three main tasks: data preprocessing (3σ outlier elimination, moving average noise reduction), initial judgment of single index threshold (response delay

controlled within 100ms), and retransmission after network disconnection. Some computing is moved to the edge, which not only reduces the pressure on the cloud but also maintains basic functionality in case of network glitches. Liu's (2025) research shows that the long-term reliability of intelligent monitoring equipment in the harsh environment of tunnels is a key guarantee for system availability.

Cloud is based on the Spring Boot+Vue.js microservice architecture and includes a data middle platform, a fusion analysis engine, an early warning engine, and a BIM engine (Three.js+WebGL lightweight rendering).

2.3 Functional Module Division

In terms of functional module splitting, Fei Guanghai et al. (2016) developed a monitoring and multi-information management system for drilling and blasting tunnels, and their modular design ideas provided a reference for this system. Wang Chao et al. (2020) studied the application of the construction integrated management platform in tunnel engineering and verified the feasibility of the platform-based data integration model. Based on this, the system was broken down into five functional modules: data acquisition and management, BIM model management, data fusion processing, safety early warning analysis, and 3D visualization display.

Specifically, the data acquisition and management module interfaces with end-layer sensor protocols (Modbus, MQTT, LoRaWAN, etc.) and is mainly used to convert heterogeneous data into an internal standard format. The BIM model management module is responsible for Revit model import and component parsing to establish a ternary mapping of "sensor ID<->BIM component ID<-> database record". The data fusion processing module, as the core of the entire system, is mainly used to perform improved DS fusion based on multiple monitoring data. The security early warning analysis module relies on a three-tier progressive early warning model to implement three-level early warning logic. The 3D visualization display module is mainly used to map the fused risk status onto the surface of the BIM model, thereby enabling real-time feedback on the system.

2.4 Data Storage Design

The system data mainly consists of two categories: field monitoring data and BIM model data. The highway tunnel construction monitoring and measurement management system developed by Li Changlong et al (2020) uses a relational database to organize monitoring data, and its storage strategy provides a reference for the design of this system. The intelligent monitoring and safety early warning system for tunnels based on blockchain structure proposed by Zhang Zhiming et al. (2019) has made beneficial explorations in the aspect of trusted data storage.

In terms of storage design, MySQL 8.0 architecture is used in the background, and the monitoring data table is partitioned by time RANGE (RANGE PARTITIONING), granularity is divided to the daily level. It is measured that the query delay is controlled at the hundred-millisecond level with millions of records. The BIM model data is stored using a "model file + metadata" strategy, where the model file is placed in object storage (MinIO), and only lightweight metadata such as file paths, version numbers, and component trees are retained in MySQL.

3. Multi-Source Monitoring Data Fusion Method

3.1 Fusion Algorithm Based on Improved DS Evidence Theory

With the continuous development of information technology, in current tunnel construction, multiple sensors are often installed at the same section. Due to the limitations of on-site construction conditions or blasting disturbances, it is difficult for each sensor to reflect the state of the surrounding rock in a consistent manner. Take the YK12+302 section of the Yingge Ling Tunnel as an example. The cumulative settlement read by the laser displacement meter is 47.2mm, while the convergence meter reading at the same section is 45.5mm. Although the trends are consistent, there are differences in absolute values, mainly due to instrument noise, differences in measurement point positions, and blasting vibration interference. When dealing with multi-source monitoring data in the current system, the Dempster-Shafer evidence theory is often used, with the core being the Dempster combination rule. Yang Linxi (2023) pointed out that when there is a high conflict among the evidence (the conflict factor K approaches 1), the traditional DS theory's Dempster fusion rule fails. Therefore, to further ensure the effectiveness of the fusion algorithm, further improvements were made based on the existing method.

(1) Confidence Hellinger distance

Let m_1 and m_2 be the BPA of two pieces of evidence, the recognition frame $\Theta=\{A_1,\ldots A_n\}$, then the confidence Hellinger distance is defined as:

$$D_H(m_1,m_2)=\frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{N}\left(\sqrt{m_1(A_i)}-\sqrt{m_2(A_i)}\right)^2} \quad (1)$$

This distance measures the difference in the "shape" of the two probability distributions, taking values [0,1]. The larger the value, the more inconsistent the two pieces of evidence.

(2) Calculation of evidence weights

First, calculate the similarity matrix: $Sim_{ij}=1-D_H(m_i,m_j)$Next, calculate support and weights:

$$Sup_i=\sum_{j=1,j\neq i}^{M} Sim_{ij},\quad \omega_i=\frac{Sup_i}{\sum_{j=1}^{M} Sup_j} \tag{2}$$

The calculation shows that the greater the conflict of evidence, the lower the support obtained from other evidence, and the smaller the weight. After on-site debugging and analysis, this paper selects that if the weight of a certain piece of evidence is less than 0.05, it is considered invalid evidence and excluded.

(3) Fusion process: Mainly through five calculation steps, namely calculating the distance matrix, converting to the similarity matrix, calculating the weights, weighted correction of BPA, and finally applying Dempster combination rules to obtain the final fusion result.

3.2 Comparative Analysis of Fusion Algorithms

To further verify the reliability of the fusion algorithm, the monitoring of the right arch top subsidence of the Yingge Ling Tunnel was selected as the data source for fusion verification. Data from the Grade V surrounding rock section (YK12+302 to YK12+308) were selected for annotation and risk classification, with a total of 30 conflict scene samples. The effects of the three fusion methods were compared and analyzed, and the results are shown in Table 1.

Table 1. Performance Comparison of Fusion Algorithms

Methods	**Fusion accuracy /%**	**Average response time /s**	**Success rate of conflict evidence handling /%**
Traditional DS evidence theory	72.3	5.2	45.6
Simple weighted average method	81.5	4.1	--
Improving DS evidence theory	92.5	2.8	96.8

The traditional DS theory has an accuracy rate of only 72.3% in conflict scenarios and a conflict resolution success rate of less than half. The simple weighted average improves to 81.5% without considering conflicts among the evidence, but there is still room for improvement. The method proposed in this paper identifies conflicts by Hellinger distance and adaptively weights them, improving accuracy to 92.5% and reducing response time from 5.2 seconds to 2.8 seconds.

3.3 Validation of Fusion Examples

Three adjacent measurement points were selected to simulate a multi-sensor scenario, taking the monitoring of arch top subsidence at the YK12+302 section of the Yingge Ling Tunnel as an example. S_1 is the G2 measurement point at YK12+302 section (cumulative settlement 47.2mm), S_2 is the G2 measurement point at YK12+308 section (cumulative settlement 46.3mm), S_3 is the G1 measurement point at YK12+308 section (cumulative settlement 45.5mm). The BPA allocation results given by the three sensors for the risk level Θ={H_{21} (normal), H_3 (warning), H (dangerous)} are shown in Table 2.

Table 2. Sensor BPA Allocation

Source	**H1**	**H2**	**H3**	**Θ**
S_1 (YK12+302 G2)	0.15	0.55	0.25	0.05
S_2 (YK12+308 G2)	0.20	0.50	0.20	0.10
S_3 (YK12+308 G1)	0.55	0.25	0.10	0.10

The results showed that the H_1 probability of S_3 was 0.55, and the data read was overly optimistic and inconsistent with S_{12} and S. From Equations 2.1 to 2.2, ω=0.$_1$372,$_2$ ω=0.373,$_3$ ω=0.255, it is obvious that the weight of S_3 is naturally reduced. After fusion, m_fusion (H_2) =0.712, judged as H_2 (warning) by the maximum membership degree. The fusion process, compared with the traditional DS (0.650) and weighted average (0.480), the improved DS evidence theory handles conflicting evidence more reasonably.

4. Security Early Warning Model

4.1 Evaluation Index System

In order to further select the early warning evaluation index system that conforms to the actual situation on site,

this paper mainly adopts the method of combining engineering investigation and questionnaire survey. Based on the "Technical Specifications for Highway Tunnel Construction" (JTG F60) and the experience of on-site engineers, and on (Wu Danhong, 2023; Liu Xianghui, 2023; He Gan, 2023; Ding Zhi, 2025) the basis of the existing research results, the factors affecting the safety of tunnel construction are summarized into four first-level dimensions:

(1) Geological environment (G, weight 0.30): surrounding rock grade, groundwater conditions, degree of development of fault zones, etc.

(2) Structural safety (S, weight 0.35): arch settlement rate, peripheral convergence rate, surrounding rock pressure, support stress, etc. This dimension has the highest weight because it directly reflects the mechanical response of the surrounding rock-support system.

(3) Construction management (M, weight 0.20): whether the excavation step distance exceeds the limit, the timeliness of support, personnel density, etc.

(4) Personnel status (P, weight 0.15): number of workers at the face, personnel positioning and distribution, abnormal physical signs, etc.

The weights are determined by the AHP-entropy weight combination weighting method: $w^{(A)}=(w^{(A)}{}^{(A)}\ldots W_j{}^{(E)}) / \Sigma (w_j{}^{(A)}\ldots w^{(E)})$. Five specialized early warning modules are set up for special disaster types in tunnel construction: collapse, gas, water gushing and mud gushing, rockburst, and large deformation.

4.2 Three-Tier Progressive Early Warning Model

Peng Zhimin (2024) conducted a systematic study on tunnel construction safety management and risk early warning technology, noting that the single threshold method is difficult to adapt to the dynamic changes of complex geological conditions. Based on this, this paper adopts a "shallow to deep" three-level early warning mechanism. The first level is a single-indicator dynamic threshold, with three levels of dynamic thresholds set for each monitoring indicator: observation value ($\mu+2\sigma$), warning value (70% of the design allowable value), and alarm value (90% of the design allowable value). The threshold range is dynamically adjusted according to different grades of surrounding rock and construction phases.

The second layer combines multi-index fusion with fuzzy comprehensive evaluation and sets five comment sets V={safe, relatively safe, average, relatively dangerous, dangerous}. When the first layer triggers a warning value or there is a multi-indicator linkage anomaly, the second layer program is entered. At this point, the multi-source evidence is fused using the improved DS evidence theory, and then the fuzzy comprehensive evaluation is carried out to further avoid the situation where other sensors do not synchronously anomaly when a single sensor misreads due to a failure.

The third layer of determination uses LSTM+Attention time series prediction. Deep prediction is initiated when the risk level of the current two-layer judgment reaches "dangerous" or "dangerous", where the LSTM network learns the temporal dependencies of the multivariable time series, and the Attention mechanism automatically focuses on the historical moments (Liu Shang, 2023) that have the greatest impact on the prediction.

4.3 Comparison Experiment of Early Warning Models

The three early warning strategies were compared using the field data of 340 arch subsidence measurement points and 464 peripheral displacement measurement points in Yingge Ling Tunnel as the dataset, and the results are shown in Table 3.

Table 3. Comparison of Performance of Early Warning Models

Early warning models	**Accuracy rate /%**	**False positive rate /%**	**Underreporting rate /%**	**Average early warning time /h**
Single threshold method	75.8	18.5	12.1	0.5
Fuzzy comprehensive evaluation method	85.2	12.3	7.8	1.2
Three-tier progressive model	92.5	8.3	3.1	4.0

From this, it can be seen that there is a problem with threshold setting in the single-threshold method. A threshold that is too low leads to frequent false alarms, while a threshold that is too high is prone to missed alarms. Although the fuzzy comprehensive evaluation method has reduced the rate of false alarms and missed alarms to some extent, there is still considerable room for improvement in the time of early warning. The model in this paper combines three methods to construct a three-level progressive model, reducing the false alarm rate from 18.5% to 8.3%, the

false alarm rate from 12.1% to 3.1%, and the early warning time from 0.5 hours to about 4.0 hours, which can provide more accurate and reliable results for real-time monitoring of engineering sites.

4.4 Early Warning Levels and Response Mechanisms

Further, on the basis of ensuring that the early warning model is more reliable, the early warning levels are classified into three levels based on actual circumstances, as shown in Table 4. The tunnel safety information early warning system developed by Liu Bin et al. (2023) verified the effectiveness of the multi-channel synchronous notification mode in engineering practice. After the warning is issued, the system can notify the corresponding responsible person in real time through multiple channels such as APP push, SMS reminder, sound and light alarm, etc., to respond and handle quickly, and ultimately achieve the closed-loop management of "warning – response – handling – feedback – cancellation".

Table 4. Warning Levels Classification

Levels	Colour	Status	Judgment conditions	Response measures
Level IV	green	Normal	Overall score ≥80	Routine inspection
Grade III	yellow	Follow	A single indicator exceeds attention or 60≤ score <80	Encrypted observations
Level II	orange	Warning	Multiple indicators abnormal or 40≤ score <60	Activate special programs and enhance support

5. Engineering Application Verification

5.1 Rely on the Project Overview

This paper is based on the Hainan Yingge Ling Extra-long Tunnel, which is located in the central mountainous area of Hainan Province. It is the longest highway tunnel under construction in Hainan Province, with a total length of 8,271 meters, a double-tunnel separated layout, a designed speed of 80km/h, and an excavation section of 87.14 meters^2. The right tunnel starts at K7+800 and ends at K12+330, with a length of 4,530 meters; The left hole runs from ZK7+800 to ZK12+352 and is 4,552 meters long.

The tunnel passes through complex strata conditions, and the surrounding rock is composed of three grades: Grade III, Grade IV and Grade V. Distribution of surrounding rock in the right tunnel: Grade III surrounding rock 2375m (52.43%), mainly composed of slightly weathered granite, with relatively intact rock mass; Grade IV rock 1310m (28.92%), mainly composed of moderately weathered sandstone and conglomerate, with well-developed joints; Grade V surrounding rock 845 m (18.65%), mainly composed of strongly weathered sandstone, conglomerate and gravel soil, with poor self-stabilizing ability. Left cave rock distribution: Grade III 2485m (54.59%), Grade IV 1300m (28.56%), Grade V 767m (16.85%). Grade V surrounding rock is concentrated at the entrance and exit of the tunnel and near the fault fracture zone, which is a key section for construction safety monitoring.

5.2 Monitoring Plan and Data Analysis

According to the geological conditions of the Yingge Ling Tunnel, the monitoring measurement items are divided into two major categories: mandatory measurement items and optional measurement items. Zhou Shui et al. (2022) practiced the application of wireless vibrating wire acquisition instruments in the monitoring and measurement of extra-long tunnels and verified the reliability of automated acquisition technology in long-distance tunnels. Mandatory items are daily monitoring and measurement items that must be carried out to ensure the stability of the surrounding environment and surrounding rock of the tunnel and construction safety, as well as to reflect the design and construction status. The specific monitoring items are shown in Table 5.

Table 5. Mandatory Monitoring and Measurement Items for Yingge Ling Tunnel

Serial Numbers	Measurement items	Methods and Tools
1	Observation inside and outside the hole	Geological compass, digital camera
2	Peripheral displacement	Total station
3	Arch sinking	Total station
4	Surface subsidence	Total station
5	Seepage pressure, water flow	Piezometer

Serial Numbers	Measurement items	Methods and Tools
6	Corrugated plate strain	Steel surface strain gauge
7	Wellbore force monitoring	Fiber grating strain gauge

In this paper, 78 surface subsidence data of the left and right tunnels (concentrated in the inlet and outlet sections of grade V surrounding rock), 340 arch top subsidence data (including 159 YK lines of the right tunnel and 181 ZK lines of the left tunnel, covering grade III to V surrounding rock), and 464 peripheral displacement data were selected as the research objects to statistically analyze the distribution of the three types of monitoring indicators. The results show that the cumulative surface subsidence values have a wide distribution range (0.1 to 48.5mm), and the median maximum deformation rate is about 1.5mm/d, indicating that the shallow-buried sections at the entrance and exit are greatly affected by surface factors. The cumulative settlement of the arch was mainly distributed in the range of 5 to 15mm, but there were several high points in the grade V surrounding rock section. Among them, the cumulative settlement of the G2 measurement point at the YK12+302 section reached 47.2mm, which was the maximum value of all measurement points. The median cumulative value of the peripheral displacement is about 4.0mm, which is relatively small overall, indicating good convergence control of the tunnel.

The deformation characteristics of the arch settlement under different surrounding rock grades were statistically analyzed. Among them, grade IV surrounding rock has the highest median deformation rate (about 2.6mm/d), which is related to the alternation of soft and hard lithology and the development of local joints in grade IV surrounding rock; The median cumulative settlement of grade V rock is the highest (about 10mm), while that of grade III is the lowest (about 5mm), because grade V rock has poor self-stabilizing ability, long deformation duration, and although the initial rate is not high, the cumulative effect is significant.

5.3 System Operation Effect

The system was deployed in the left and right tunnels of Yingge Ling Tunnel and operated continuously for 6 months. The statistics of the early warning effect are shown in Table 6.

Table 6. System Early Warning Effect Statistics

Indicators	Numerical
Warning accuracy	92.5%
Average response time	2.8 seconds
False alarm rate	8.3%
Underreporting rate	3.1%
System availability	99.2%

Among them, at the G2 measurement point of the YK12+302 section (right tunnel K12+302, grade V surrounding rock section), the system detected a cumulative settlement of 47.2mm and a deformation rate of 4.2mm/d at 14:32, exceeding the warning threshold (3.0mm/d), triggering a yellow alert for the first layer; At 14:35, the second layer was upgraded to an orange alert after merging with the adjacent G1 measurement point at the YK12+308 section; At 14:38 The third layer LSTM predicts that the alarm value will be reached within the next 4 hours. The site promptly took measures such as adding steel supports and shortening excavation footage, and the deformation rate gradually dropped back to 2.1mm/d. This also validates the rationality of the high weight of the “structural safety” dimension in the “4+4+N” index system.

References

Ding Zhi. (2025). Research on Safety Risk Management in the Construction Process of tunnel Engineering projects. *Transport Manager World*, (14), 70-72.

Fei Guanghai, Wu Xiaoping, Li Wenrong, et al. (2016). Drilling and blasting tunnel construction monitoring and multi-information management system. *Journal of Railway Science and Engineering*, *13*(4), 775-782.

He Gan. (2023). Research on Construction Technology and Safety Early Warning Standards for the Entrance Section of Four-lane Additional Tunnels. Chongqing Jiaotong University.

He Xuji. (2022). Application of safety management and risk warning technology in tunnel construction. *Engineering Construction & Design*, (5), 222-224.

Li Changlong, Lu Fengwen, Ji Tongxu. (2020). Development of monitoring and measurement management system

for highway tunnel construction. *Journal of Jiamusi University: Natural Science Edition*, *38*(5), 5.

Liu Bin, Cui Xian, Sun Guohui, et al. (2023). Research and application of tunnel safety information Early warning system. *Industrial Building*, *53*(Supplement), 899-903.

Liu Decai. (2025). Research on Safety Risk Early Warning Mechanism for Tunnel Construction Based on Intelligent Equipment. *Smart Building & Smart City*, (5), 152-154.

Liu Shang. (2023). Dynamic Assessment of Safety Risks throughout Tunnel Construction Based on Data Mining and its Engineering application. Shandong University.

Liu Xianghui. (2023). Research on Feature Extraction and Early Warning Application of Construction Safety Accidents in Mountain Tunnels. Changsha University of Science & Technology.

Liu Xinze. (2024). Automation monitoring technology in safety early warning of Tunnel construction. *Construction Machinery & Maintenance*, (1), 70-72.

Ma Haowei, Xiao Lina. (2025). Application of UWB technology in Tunnel Positioning Management. *Equipment Management & Maintenance*, (12), 116-118.

Ministry of Transport of the People's Republic of China. (2024). Statistical Bulletin on the Development of the Transport Industry 2024.

Peng Zhimin. (2024). Research on Safety Management and Risk Warning Technology for Tunnel construction. *Engineering Construction & Design*, (24), 220-222.

Qiao Xiong, Hu Shijing, Tian Zheng. (Year). Research progress on the application of real-time automatic monitoring technology during tunnel construction. *Tunnel Construction (Chinese and English)*, *45*(1), 1-15.

Song Xiuguang, Tian Weiyang, Wei Mingzhao, et al. (2026). Tunnel structure health monitoring technology research status and prospect. *Journal of shandong university (engineering science)*, 1-17. https://link.cnki.net/urlid/37.1391.T.20260510.1652.002.

Wang Chao, Zhou Leisheng, Xu Run, et al. (2020). Application research of construction integrated management platform in tunnel engineering. *Journal of Chongqing Jiaotong University (Natural Science Edition)*, *39*(9), 74-79.

Wang Yaqiong, Huang Yilin, Wang Kaiyun, et al. (2015). Application of Internet of Things Based on Wireless Sensing in tunnel construction monitoring. *Journal of Xi'an University of Science and Technology*, *35*(4), 498-504.

Wu Danhong. (2023). Design of Risk Assessment and Safety Early Warning System for Expansion Construction around Existing Tunnels. China University of Geosciences.

Yang Linxi. (2023). Intelligent Evaluation and real-time Warning of Safety Risks in Offshore Tunnel Construction Based on BIM. Chongqing: Chongqing Jiaotong University.

Zhang Jiasong, Huang Zhou, Leng Zhiming. (2021). Application of Automated Monitoring and early warning technology in highway tunnel engineering. *Highway*, *66*(6), 416-418.

Zhang Junjia, Leng Xianlun, Sheng Qian, et al. (2019). Application of Automated Monitoring technology in Safety Early warning for Tunnel Construction. *Highway Engineering*, *44*(2), 14-18.

Zhang Zhiming, Ye Ying. (2019). Intelligent monitoring and safety warning system for tunnels based on Blockchain structure. *Modern Tunnel Technology*, *56*(S2), 73-79.

Zhao Jinna, Hu Xuefei. (2026). Research on Dynamic Monitoring Strategies for Railway Tunnel Construction Safety Empowered by BIM-GIS Technology. *Engineering Construction & Design*, (6), 105-108.

Zhou Shui, Luo Yi, Ye Baoquan, et al. (2022). Application and Practice of wireless vibrating wire Acquisition in Monitoring and measurement of Extra-long tunnels. *Transport Manager World*, (4), 106-108.

Zhou Yefei, Wang Peng, Xu Linbo, et al. (2026). Research on the Construction of a "Digital shield" system and the application of Collaborative Early Warning Mechanism for Dabie Mountain Tunnel Construction Safety. *China Highway*, (2), 99-101.

Paradigm Academic Press
Innovation in Science and Technology
ISSN 2788-7030
JUN. 2026 VOL.5, NO.2

A Survey of Classic Machine Learning Algorithms: Principles and Applications

Xi Long[1] & Xing Zhao[2]

[1] School of Statistics and Data Science, Capital University of Economics and Business, Beijing 100071, China

[2] School of Urban Economics and Public Administration, Capital University of Economics and Business, Beijing 100071, China

Correspondence: Xing Zhao, School of Urban Economics and Public Administration, Capital University of Economics and Business, Beijing 100071, China.

doi:10.63593/IST.2788-7030.2026.06.003

Abstract

Classical machine learning algorithms constitute the fundamental cornerstone of modern data science and intelligent system development. While deep learning has achieved transformative breakthroughs across numerous fields in recent years, classical methods remain indispensable in practical scenarios characterized by limited training data, stringent interpretability requirements, or constrained computational resources. Nevertheless, existing studies generally lack a systematic, unified, and beginner-friendly comprehensive survey that integrates theoretical elaboration, multi-dimensional comparative analysis, and actionable algorithm selection guidance.

To fill this research gap, this paper presents a thorough investigation of ten representative classical machine learning algorithms: Linear Regression, Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, K-Nearest Neighbors, Naive Bayes, Adaptive Boosting, K-Means, and Principal Component Analysis. Each algorithm is explicated within a consistent structural framework, covering its mathematical formulation, core working mechanism, distinct advantages, inherent limitations, and typical application scenarios. Furthermore, a horizontal comparative analysis is conducted across seven critical dimensions, and a practical algorithm selection framework with targeted recommendations for typical industrial scenarios is proposed.

This work constructs a complete and logically coherent knowledge system of classical machine learning, serving as an accessible and pragmatic reference for novice learners and engineering practitioners. It also provides insights into future integration trends between classical algorithms and emerging technologies including large language models, explainable artificial intelligence, edge intelligence, automated machine learning, and federated learning.

Keywords: machine learning, algorithm survey, comparative analysis, algorithm selection, mathematical formulation

1. Introduction

1.1 Background and Definition

Machine learning (ML) is a fundamental subfield of artificial intelligence (AI) that enables computer systems to automatically learn and improve from experience without being explicitly programmed to perform specific tasks. (Michalski, 2013)

The most rigorous and widely cited definition of machine learning was provided by Mitchell in his landmark textbook: “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E”. (Mitchell, 1997) This definition elegantly captures the three essential components of any machine learning system: (1) Task

(T): the specific problem the system is designed to solve, such as classification, regression, or clustering; (2) Experience (E): the data used to train the system; and (3) Performance Measure (P): a quantitative metric used to evaluate how well the system performs the task.

This framework provides a unified lens through which we can understand all the classic machine learning algorithms discussed in this survey, as each algorithm represents a different approach to optimizing performance P given experience E for a particular class of tasks T.

1.2 Historical Evolution of Machine Learning

The evolution of machine learning over the past 70 years has been defined by alternating periods of rapid theoretical progress and pragmatic retrenchment, driven by the interplay between computational capabilities, mathematical insights, and real-world demand. The field traces its intellectual origins to the 1950s, when early AI researchers first proposed that machines could be designed to learn from data rather than relying solely on hand-coded rules. The seminal work of Samuel on a self-learning checkers program provided the first empirical demonstration of this concept, showing that a computer could outperform its human programmer by iteratively improving its strategy through self-play. (Samuel, 1959) This early era was dominated by symbolic approaches and optimistic claims about general AI, but these efforts ultimately stalled in the 1970s due to fundamental limitations in computational power and the inability to handle complex, noisy real-world data—a period now known as the first “AI winter”. (Toosi, 2021)

The 1980s and 1990s marked a decisive shift toward statistical learning theory, which remains the mathematical foundation of modern classical machine learning. (Wu, 1999) During this period, researchers developed rigorous theoretical frameworks for understanding generalization, overfitting, and model complexity, most notably Vapnik’s statistical learning theory, which laid the groundwork for support vector machines (SVMs) and established formal bounds on model performance. (Vapnik, 1999) This era also saw the maturation of many core algorithms that remain widely used today, including decision trees, k-nearest neighbors, and the backpropagation algorithm for neural networks. By the late 1990s, machine learning had emerged as a distinct discipline separate from general AI, with a focus on solving practical, data-driven problems.

The 2000s witnessed the rise of ensemble learning methods and the widespread adoption of machine learning in industrial applications. Algorithms such as random forests and gradient boosting machines combined multiple weak learners to achieve state-of-the-art performance on a wide range of tasks, while advances in computing infrastructure made it possible to process increasingly large datasets. During this period, classical machine learning became the dominant approach for data analysis across industries, from finance and healthcare to marketing and cybersecurity.

The past decade has been dominated by the explosive growth of deep learning, which has achieved remarkable success in areas such as computer vision, natural language processing, and speech recognition. (Zhou, 2021) This has led some to question the continued relevance of classical machine learning algorithms. However, classical methods remain indispensable in many real-world scenarios, particularly when data is limited, interpretability is critical, or computational resources are constrained. Moreover, a solid understanding of classical algorithms is essential for grasping the fundamental principles that underpin all machine learning, including deep learning. (Ezugwu, 2024)

1.3 Related Work and Research Gap

Despite the vast body of literature on machine learning, there remains a critical gap in accessible, comprehensive introductions to the core classical algorithms for beginners. Existing surveys tend to fall into one of three categories: highly theoretical treatises that are inaccessible to non-experts, domain-specific reviews that focus on particular applications rather than general algorithms, or broad overviews that sacrifice depth for breadth. Few studies provide a systematic, unified treatment of the ten most fundamental algorithms, covering their mathematical principles, advantages, limitations, and practical applications in a consistent format. Furthermore, there is a lack of clear, actionable guidance for selecting the appropriate algorithm for a given problem—a common pain point for novice practitioners.

1.4 Contributions of This Paper

This paper addresses these gaps by presenting a comprehensive, beginner-friendly survey of ten classic machine learning algorithms that form the foundation of modern data science. The contributions of this work are threefold: (1) A systematic introduction to the core principles and mathematical formulations of each algorithm in a unified framework; (2) A rigorous comparative analysis of their strengths, weaknesses, and applicable scenarios; (3) A practical algorithm selection guide to help practitioners make informed decisions.

2. Fundamental Concepts of Machine Learning

2.1 Major Learning Paradigms

Machine learning algorithms are primarily categorized into two fundamental paradigms based on the type of training data and learning objectives.

Supervised learning is the most widely used paradigm, where algorithms learn from labeled datasets. (Liu, 2011) Each sample in the training data consists of input features paired with a corresponding output label. The algorithm learns a mapping function from features to labels, enabling it to predict labels for new, unseen samples. Supervised learning addresses two core task types:

(1) Classification: Predicting discrete categorical labels, such as identifying whether an email is spam or legitimate.

(2) Regression: Predicting continuous numerical values, such as forecasting monthly sales or estimating house prices.

Unsupervised learning operates on unlabeled datasets, where no predefined output labels are provided. (Barlow, 1989) The goal is to discover hidden patterns, structures, or inherent relationships within the data. The two primary unsupervised learning tasks are:

Clustering: Grouping similar samples into distinct clusters based on their feature similarity.

Dimensionality reduction: Reducing the number of input features while preserving the most critical information in the dataset.

Other advanced learning paradigms, including semi-supervised learning and reinforcement learning, are beyond the scope of this survey.

2.2 Standard Machine Learning Workflow

A typical machine learning project follows a standardized five-stage workflow that ensures systematic development and deployment of reliable models (Zhou, 2021):

Data collection and preprocessing: Gathering relevant data and cleaning it to handle missing values, outliers, and inconsistent formats.

Feature engineering: Transforming raw data into meaningful features that better represent the underlying problem to the model.

Model selection and training: Choosing an appropriate algorithm based on the problem type and data characteristics, then training it on the training dataset to learn the feature-label mapping.

Model evaluation and tuning: Assessing model performance on a separate validation dataset and adjusting hyperparameters to optimize generalization ability.

Model deployment and monitoring: Deploying the trained model to production environments and continuously monitoring its performance over time to detect degradation.

2.3 Brief Overview of Performance Measure

Evaluating the generalization performance of a learner requires not only effective and feasible experimental estimation methods but also evaluation criteria to measure the generalization ability of the model, which is referred to as performance measure. Quantitative performance measures are essential for objectively evaluating and comparing different models.

The choice of metric depends on the specific task type:

The evaluation metrics for regression tasks focus on the difference between the predicted value and the actual value. For regression tasks, the most commonly used measures are:

(1) Mean Squared Error (MSE): Measures the average squared difference between predicted and actual values, penalizing larger errors more heavily. (Fahrmeir, 2022)

Given a dataset D containing n samples. Given a sample set, where y is the true label of example x_i. To evaluate the performance of learner f, we need to compare the learner's prediction $f(x)$ with the true label y. The most commonly used performance metric for regression tasks is Mean Squared Error.

$$E(f;D) = \frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2$$

(2) Mean Absolute Error (MAE): Measures the average absolute difference between predicted and actual values, providing a more robust measure of error magnitude. (Fahrmeir, 2022)

For classification tasks, let's illustrate with a simple binary classification example. The first concept to clarify is that the system needs to classify samples into two categories: positive and negative. (Rupp, 2008)

The box in the figure below represents all the samples that need to be predicted. The samples within the blue circle represent those that the system predicts as positive, while the samples within the green circle represent those that are actually positive.

So the part where blue and green intersect represents True Positive (TP), where the system predicts true and the actual situation is also true, indicating that the system's prediction is correct (True);

The green part outside the blue circle represents samples that the system predicted as false but are actually true. These are False Negatives (FN), where the system made an error in prediction (False);

The blue part outside the green circle represents samples that the system predicts as true but are actually false. These are False Positives (FP), where the system made an error in prediction;

Finally, the area outside the two circles represents samples that the system predicts as negative and are indeed negative, namely, correctly predicted True Negatives (TN).

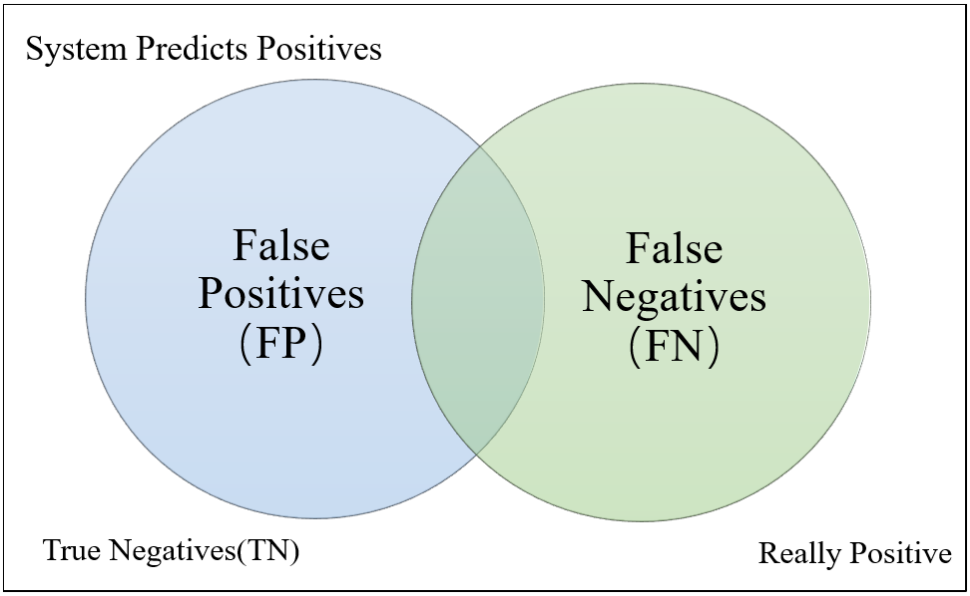

Figure 1.

The above figure can be summarized by the following table, which is called a confusion matrix.

Table 1.

	YES	NO
YES	TP	FN
NO	FP	TN

For binary classification problems, the confusion matrix is a 2×2 table, where the rows represent the true labels of the samples and the columns represent the predicted labels of the model. The first row indicates samples that are truly positive, while the second row indicates samples that are truly negative; the first column indicates samples that are predicted as positive, while the second column indicates samples that are predicted as negative.

It can be seen that the elements on the main diagonal (TP and TN) correspond to correctly predicted samples, while the elements on the subdiagonal (FN and FP) correspond to incorrectly predicted samples. Therefore, a classifier with good performance should have a larger value on the main diagonal and a smaller value on the subdiagonal.

The standard measures include:

(1) Error Rate and Accuracy: The ratio of the number of misclassified samples to the total number of samples. The overall proportion of correct predictions, suitable for balanced datasets.

$$ErrorRate = \frac{error}{total} = \frac{FP + FN}{TP + TN + FP + FN}$$

$$Accuracy = (1 - error) = \frac{correct}{total} = \frac{TP + TN}{TP + TN + FP + FN}$$

(2) Precision and Recall: Provide detailed insights into performance on specific classes, critical for imbalanced datasets where misclassification costs vary.

Precision(P):

$$P = \frac{TP}{TP + FP}$$

Recall(R):

$$R = \frac{TP}{TP + FN}$$

(3) F-score: Other fields use different criteria. For example, F-score is commonly used in information retrieval and represents the harmonic mean of Recall and Precision, which are combined as follows:

$$F_\beta = \frac{(\beta^2 + 1) \cdot P \cdot R}{\beta^2 P + R}$$

The F-score has a parameter β, which assigns different weights to Recall and Precision by adjusting this parameter. When this parameter is set to 1, it becomes F_1, indicating that Recall and Precision have equal weights.

$$F_1 = \frac{2PR}{P + R}$$

It is easy to see that when the value of Recall or Precision is small, the F-score will also be small. Therefore, regardless of how high one of these values is, as long as the other is small, the F-score will be small.

3. Ten Classic Machine Learning Algorithms

Next, we will introduce ten classic machine learning algorithms, and explain the basic principles and applications of each algorithm.

3.1 Linear Regression Algorithms

Linear regression is a fundamental predictive method in machine learning. It is based on the assumption that there is a linear relationship between the input (features) and the output (target).

The core principle of linear regression is to find a set of weights (coefficients) that make the linear combination of these weights and features as close as possible to the target value. During the training process, these weights are determined by minimizing the difference between the predicted values and the actual values, which is usually the sum of squared errors.

In regression analysis, if only one independent variable and one dependent variable are included, and the relationship between them can be approximately represented by a straight line, then this type of regression analysis is called simple linear regression analysis. (Montgomery, 2021) It usually takes the form of:

$$y = \beta_0 + \beta_1 x + \varepsilon$$

Where y is the target variable, x is the feature, β_0 is the intercept, β_1 is the slope, and ε is the error term.

If regression analysis involves two or more independent variables, and there is a linear relationship between the dependent variable and the independent variables, it is referred to as multiple linear regression analysis. The general form is as follows:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \varepsilon$$

In this formula, $x_1, x_2, \ldots x_n$ represent the features, while $\beta_0, \beta_1, \beta_2, \ldots \beta_n$ are the coefficients of these features.

The linear regression model possesses the following advantages:

(1) Simplicity and intuitiveness: The linear regression model is easy to understand and implement, making it a suitable learning tool for beginners in machine learning.

(2) High interpretability: The results of a linear regression model are easy to interpret, clearly showing how features affect the prediction.

(3) Fundamental status: As a fundamental model in machine learning, it serves as a cornerstone for understanding more complex models.

(4) Computational efficiency: Compared to more complex models, linear regression models have high computational efficiency and are easy to handle large-scale data processing.

Although the linear regression model is a relatively basic algorithm, it plays a significant role in our lives. In cases where there is a clear linear relationship between features and the target variable, linear regression serves as a powerful predictive tool. For instance, it can be applied to predicting housing prices and consumer spending. Additionally, linear regression can be used to analyze the impact of different features on the target variable, such as analyzing the effects of various factors on outcomes in economics and social sciences.

Although linear regression performs well in many scenarios, its performance may not be ideal when dealing with nonlinear relationships, highly complex datasets, or data with a significant amount of noise. Therefore, it is crucial to understand the nature and requirements of the data before selecting a linear regression model.

3.2 Logistic Regression Algorithms

Logistic regression is a generalized linear regression analysis model that estimates the probability of an event occurring based on a given set of independent variable data. Since the result is a probability, the range of the dependent variable is between 0 and 1.

The core principle of logistic regression is to utilize a logistic function to map the output of a linear regression model to a value between 0 and 1. This mapping enables logistic regression to estimate the probability of an event occurring and perform classification accordingly.

In logistic regression problems, we typically employ the Sigmoid function model, which takes the form of (Zou, 2019):

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Here, Z is typically a linear function, such as $Z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$, where $\beta_0, \beta_1, \beta_2 \ldots \beta_n$ are the model parameters, and $x_1, x_2, \ldots x_n$ are the features.

Based on this, we can construct a logistic regression model, which takes the form of:

$$P(Y = 1|X) = \sigma(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n)$$

Where $P(Y = 1|X)$ represents the probability of the target variable Y being 1 given the feature X.

The logistic regression model possesses the following advantages:

(1) Probabilistic interpretation: Logistic regression not only provides classification results but also gives the probability of belonging to a certain category, which is very useful in many applications.

(2) Computational efficiency: Logistic regression typically boasts high computational efficiency, particularly when dealing with large datasets.

(3) High interpretability: Compared to some complex machine learning models, the results of logistic regression are easier to interpret.

(4) Broad applicability: Logistic regression can be extended to multi-class classification problems and has applications in many different fields.

Logistic regression models are commonly used to handle binary classification problems, such as spam detection, disease diagnosis, credit application approval, etc. Logistic regression can also be used to estimate probabilities, making it highly applicable in scenarios where the probability of an event needs to be estimated, such as banks calculating customer default probabilities. Furthermore, logistic regression performs well when there is a linear relationship between features and classification outcomes.

Although logistic regression performs well in the aforementioned problems, its effectiveness may be compromised when dealing with nonlinear relationships, complex interactions between features, or very high-dimensional data. In these cases, more complex models such as Random Forests, Support Vector Machines, or Neural networks may need to be considered.

3.3 Decision Tree Algorithms

Decision tree is a common type of machine learning method. Since the decision branches are drawn like the branches of a tree, it is called a decision tree. (Zhou, 2021) In machine learning, a decision tree is a predictive model that represents a mapping relationship between object attributes and object values.

The original decision tree algorithm, known as the Concept Learning System (CLS), was proposed by psychologist

and computer scientist E. B. Hunt in 1962 during his research on human conceptual learning processes. This algorithm established the "divide and conquer" learning strategy for decision trees. (Hunt, 1962)

The core principle of a decision tree is to decompose a complex decision-making process into a series of simpler decisions, thereby forming a tree-like structure. When constructing a decision tree, the algorithm selects the optimal feature for splitting, so as to clearly divide the dataset at each node. This process is repeated until a certain stopping criterion is met, such as the maximum depth of the tree or the minimum number of samples in a node.

It recursively partitions the feature space into mutually exclusive rectangular regions, with each region corresponding to a final prediction result. A decision tree consists of three types of nodes: the root node (representing the entire dataset), internal nodes (representing feature tests), and leaf nodes (representing final predictions). The core learning logic of the algorithm is to select the optimal feature for splitting at each node, maximizing the purity of the split child nodes, that is, ensuring that the samples within the same node belong to the same category as much as possible. (Li, 2024)

The mainstream decision tree algorithm employs three core splitting criteria:

(1) Entropy and Information Gain (ID3 Algorithm)

Entropy is used to measure the purity or uncertainty of a dataset. (Hssina, 2014) For a dataset D containing K categories, the definition of entropy is:

$$H(D) = -\sum_{i=1}^{K} p_i \, log_2(p_i)$$

Where p_i denotes the proportion of the i-th category of samples in the set D.

Information gain (IG) measures the decrease in entropy of the dataset after splitting based on feature A:

$$IG(D, A) = H(D) - H(D|A)$$

The ID3 algorithm selects the feature with the highest information gain for splitting.

(2) Information Gain Ratio (C4.5 Algorithm)

To address the issue of ID3's bias towards features with a large number of values, C4.5 employs the information gain ratio to normalize the information gain based on the inherent value of the feature (Hssina, 2014):

$$IGR(D, A) = \frac{IG(D, A)}{H_A(D)}$$

Where $H_A(D)$ represents the entropy of the dataset with respect to the values of feature A.

(3) Gini coefficient (CART algorithm)

Classification and Regression Tree (CART) uses the Gini coefficient to measure the probability of a randomly selected sample being misclassified:

$$Gini(D) = 1 - \sum_{i=1}^{K} p_i^2$$

CART is a binary tree algorithm that can handle both classification and regression tasks simultaneously. To prevent overfitting, decision trees often employ pruning techniques (either pre-pruning or post-pruning) to remove unnecessary branches. (Chipman, 1998)

The decision tree model possesses the following advantages:

(1) High interpretability: The results of decision trees are easy to understand and interpret, and even non-professionals can grasp the decision-making path.

(2) No need for data standardization: Decision trees do not have strict requirements for data standardization or normalization, unlike some other algorithms.

(3) Can handle nonlinear relationships: Decision trees can effectively handle nonlinear relationships between data.

(4) Can handle mixed-type data: Decision trees can handle both numerical and categorical features simultaneously.

Decision trees perform well in handling binary or multi-class classification problems and regression tasks involving continuous value prediction. Additionally, during its construction, a decision tree selects the most important features, making it suitable for feature selection. Furthermore, for datasets with complex relationships between features, decision trees can capture these relationships as well.

Despite the numerous advantages of decision trees, they are prone to overfitting, especially when the tree becomes deep or complex. To prevent overfitting, pruning or limiting the depth of the tree is often necessary. Furthermore,

ensemble methods based on decision trees, such as random forests and gradient boosting trees, can further enhance the performance and generalization ability of the model.

3.4 Random Forest Algorithms

Random Forest (RF) belongs to the category of ensemble learning. It improves the accuracy and stability of predictions by combining multiple decision trees.

Leo Breiman is the primary proponent and founder of the Random Forest algorithm. Together with Adele Cutler, he refined and popularized this algorithm, making it one of the most classic and widely used ensemble learning models in the field of machine learning (Breiman, 2001).

RF is an extended variant of Bagging. On the basis of constructing a Bagging ensemble with decision trees as base learners, RF further introduces random attribute selection during the training process of decision trees. Specifically, when selecting a splitting attribute, a traditional decision tree chooses the optimal attribute from the attribute set (assuming there are d attributes) of the current node; In RF, for each node of the base decision tree, a subset containing k attributes is randomly selected from the attribute set of that node, and then an optimal attribute is chosen from this subset for partitioning. The parameter k here controls the degree of randomness introduced: If we let $k = d$, the construction of a decision tree based on rules is the same as that of a traditional decision tree; If we $k = 1$, then a random attribute is selected for partitioning. Generally, the recommended value is $k = log_2$ d (Breiman, 2001).

Random forest effectively reduces the correlation between base learners through randomness at two levels: sample randomness (bootstrap sampling) and feature randomness (random attribute selection), thereby significantly enhancing the generalization ability of the ensemble model. The following provides a brief introduction to these two aspects.

(1) Bootstrap Sampling

Each base decision tree in a random forest is trained based on an independent bootstrap sampling dataset. For the original dataset D containing N samples, bootstrap sampling generates a new training set D_i by randomly drawing N samples with replacement.

In the process of bootstrap sampling, the probability that a sample remains unselected throughout N rounds of drawing is:

$$\lim_{N\to\infty}(1-\frac{1}{N})^N = \frac{1}{e} \approx 0.368$$

This means that the training set of each base decision tree contains approximately 63.2% of the samples from the original dataset, and the remaining 36.8% of samples, known as Out-of-Bag (OOB) samples, can be used to estimate the model's generalization error.

(2) Integrated prediction mechanism

For classification tasks, the majority voting method is adopted, where the category with the most votes among the prediction results of all base decision trees is taken as the final prediction result:

$$f(x) = \text{argmax}_{c\in C}\Sigma_{i=1}^{T} I(h_i(x) = c)$$

Where T is the number of decision trees, $h_i(x)$ is the prediction of the i-th decision tree for sample x, $I(\cdot)$ is an indicator function, and C is the set of all categories.

For regression tasks, the simple average method is adopted, where the average of the prediction results from all base decision trees is taken as the final prediction result:

$$f(x) = \frac{1}{T}\Sigma_{i=1}^{T} h_i(x)$$

(3) Out-of-Bag Error (OOB Error)

Random forests do not require an additional validation set and can directly estimate the generalization error through out-of-bag samples. For each sample x_j , all base decision trees that do not use it as a training sample are used for prediction, and then the error rate of these predictions is calculated, which is the out-of-bag error:

$$\text{OOB Error} = \frac{1}{N}\sum_{j=1}^{N} I(f_{OOB}(x_j) \neq y_j)$$

Where $f_{OOB}(x_j)$ is the prediction result of x_j based on out−of−bag samples, and y_j is the true label of x_j.

Random forest has the following advantages:

(1) Strong generalization ability: By integrating multiple decision trees, the problem of overfitting in a single decision tree is effectively solved, and the generalization performance is significantly better than that of a single decision tree;

(2) High robustness: insensitive to outliers and noisy data, and stable performance on complex datasets;

(3) Low requirements for data preprocessing: It inherits the advantages of decision trees, eliminating the need for data normalization or standardization;

(4) Parallel computation: The training processes of individual base decision trees are completely independent, allowing for efficient parallelization and making them suitable for handling large-scale datasets;

(5) Built-in feature importance evaluation: It can automatically calculate the contribution of each feature to the prediction results.

Random forest is one of the most commonly used machine learning algorithms in the industry, widely applied to various classification and regression tasks, such as customer churn prediction, credit scoring, disease diagnosis, image classification, and recommendation systems. Random forest excels in various classification and regression tasks. At the same time, due to its efficiency, random forest is also suitable for handling large-scale datasets. Similarly, random forest can identify important features, which is very effective for feature selection. In addition, random forest can handle complex nonlinear relationships between features.

The main drawback of Random Forest is that the model may become relatively complex, which may result in less interpretability compared to a single Decision Tree. Furthermore, although Random Forest typically has a fast training speed, the required computational resources may increase when dealing with extremely large datasets. Nevertheless, Random Forest remains a highly powerful and popular algorithm in machine learning.

3.5 Support Vector Machine Algorithms

Vladimir Naumovich Vapnik and Alexey Chervonenkis jointly developed the Vapnik-Chervonenkis theory between 1960 and 1990, laying the foundation for statistical learning theory. He is the primary inventor of the Support Vector Machine (SVM), having first proposed the original algorithm in 1964, and in the 1990s, he and his colleagues introduced the kernel trick and soft-margin SVM. (Vapnik, 1999)

For a given training sample set $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}, y_i \in \{-1, +1\}$, the fundamental idea of classification learning is to find a dividing hyperplane in the sample space based on the training set D, separating samples of different categories. (Zhou,2021) The core idea of Support Vector Machine (SVM) is to seek an optimal hyperplane in the feature space, separating samples of different categories, and maximizing the minimum distance (i.e., the margin) from the hyperplane to the samples of both categories.

The core advantage of SVM lies in its ability to maximize the margin, thereby ensuring the model's generalization ability, even on small sample datasets. Based on the linear separability of data, SVM is mainly divided into three categories: linearly separable SVM (hard margin), linear SVM (soft margin), and nonlinear SVM (kernel trick). This article focuses on the most basic linearly separable SVM and briefly explains the other two extended forms:

(1) Linearly separable SVM (hard margin)

For a linearly separable binary classification dataset, there exist an infinite number of hyperplanes that can separate the two classes of samples. However, SVM selects the hyperplane with the largest margin. The mathematical expression for the hyperplane is:

$$w^T x + b = 0$$

Where $w = (w_1; w_2; \dots; w_d)$ is the normal vector, determining the direction of the hyperplane; And b is the displacement term, determining the distance between the hyperplane and the origin. Clearly, the dividing hyperplane can be determined by the normal vector w and the displacement b, which we denote as (w, b) below, The distance r from any point x in the sample space to the hyperplane (w, b) can be written as:

$$r = \frac{|w^T x + b|}{\|w\|}$$

The optimization objective of SVM is to maximize the margin $\frac{2}{\|w\|}$, which is equivalent to minimizing $\frac{1}{2}\|w\|^2$, while satisfying the constraint conditions:

$$y_i(w^T x_i + b) \geq 1, i = 1,2, \dots, N$$

Where $y_i \in \{-1,1\}$ is the label of the sample.

(2) Extended form

The first extension is the soft-margin SVM, which is designed to address linearly inseparable problems. It

introduces slack variables $\xi_i \geq 0$, allowing some samples to violate the margin constraint, while controlling the degree of penalty through the penalty parameter C.

The second extension form is nonlinear SVM, which maps low-dimensional linearly inseparable data to a high-dimensional feature space through a kernel function, making it linearly separable in the high-dimensional space. Commonly used kernel functions include linear kernel, polynomial kernel, and Gaussian kernel (RBF):

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$$

Where $\phi(\cdot)$ is a mapping function, and the kernel function avoids explicit computation of high-dimensional mappings, greatly reducing computational complexity.

SVM has the following advantages:

(1) Effectiveness: In high-dimensional spaces, especially when the boundaries between categories are clear, SVM performs very well.

(2) Strong generalization ability: SVM attempts to maximize the margin, thus it usually has good generalization ability.

(3) Scalability: By selecting an appropriate kernel function, SVM can effectively handle nonlinear problems.

(4) Strong adaptability: SVM can be used to solve various types of machine learning problems.

Based on these advantages, SVM has a wide range of applications. Initially designed to solve binary classification problems, especially when dealing with high-dimensional data, SVM can effectively handle nonlinear data through the use of kernel tricks. It performs exceptionally well when dealing with high-dimensional data such as text and images. Furthermore, although SVM is primarily used for classification, with appropriate modifications, it can also be used for regression (known as Support Vector Regression, SVR).

One of the main drawbacks of SVM is its low computational efficiency for large-scale datasets, as its training process requires a long time.

In addition, selecting appropriate kernel functions and parameters (such as regularization parameters and kernel parameters) is crucial for achieving good performance, but this often requires professional knowledge and experimental tuning.

3.6 K-Nearest Neighbor Algorithms

The k-nearest neighbor algorithm is one of the most intuitive and easily understandable machine learning algorithms, belonging to the categories of instance-based learning and lazy learning. It was originally proposed by Cover and Hart. (Cover, 1968) Unlike most algorithms that require training the model before making predictions, kNN does not have an explicit training process. It stores all training data directly and performs calculations only during prediction. kNN is simple and straightforward: given a training dataset, for a new input instance, find the k nearest instances to it in the training dataset. If the majority of these k instances belong to a certain class, then classify the input instance into that class. (Li, 2021)

Typically, “Majority Voting” can be employed in classification tasks, which involves selecting the category label that appears most frequently among the k samples as the prediction result; In regression tasks, the “average method” can be employed, which involves using the average of the real-valued output labels of these k samples as the prediction result. Alternatively, weighted average or weighted voting can be conducted based on the distance, with samples closer to the query having a higher weight.

The core of kNN lies in calculating the “similarity” between samples, with the most commonly used similarity measurement method being the distance function. Below, we introduce three of the most frequently used distance formulas:

(1) Euclidean Distance: This is the most commonly used distance metric, used to calculate the linear distance between two samples in an n-dimensional space. The mathematical formula is:

$$\mathrm{d}(x_i, x_j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2}$$

Where $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$ and $x_j = (x_{j1}, x_{j2}, \dots, x_{jn})$ are two n-dimensionalsample vectors.

(2) Manhattan Distance: Manhattan distance, also known as city block distance, is used to calculate the sum of the absolute distances between two samples in each dimension. The mathematical formula is:

$$d(x_i, x_j) = \sum_{k=1}^{n} |x_{ik} - x_{jk}|$$

(3) Minkowski Distance: The general form of Euclidean distance and Manhattan distance:

$$d(x_i, x_j) = \left(\sum_{k=1}^{n} |x_{ik} - x_{jk}|^p\right)^{\frac{1}{p}}$$

When $p = 1$, it degenerates to the Manhattan distance; when $p = 2$, it degenerates to the Euclidean distance.

When facing different task types, different methods need to be adopted. Below, we introduce the selection of methods for classification tasks and regression tasks respectively:

(1) Classification problem:

There are primarily two methods for classification problems:

The first is the majority voting method, which is the most basic classification rule. The prediction result is the category with the highest frequency among the K nearest neighbors. The mathematical formula is expressed as:

$$\hat{y} = argmax_{c \in C} \sum_{i=1}^{K} I(y_i = c)$$

Where C is the set of all categories, and $I(\cdot)$ is an indicator function that takes the value of 1 when the condition inside the parentheses holds, and 0 otherwise.

The second method is Weighted Voting. To reflect that samples closer in distance have higher importance, weights can be assigned based on the reciprocal of the distance. The mathematical formula is expressed as:

$$\hat{y} = \text{argmax}_{c \in C} \sum_{i=1}^{K} \frac{1}{\text{d}(x, x_i)^2} I(y_i = c)$$

(2) Regression problem:

There are primarily two methods for regression problems:

The first method is the average method, where the predicted result is the arithmetic mean of the values of K nearest neighbor samples. The mathematical formula is:

$$\hat{y} = \frac{1}{k} \Sigma_{i=1}^{k} y_i$$

The second method is the weighted average method, where we can also apply weights based on distance:

$$\hat{y} = \frac{\sum_{i=1}^{k} \frac{1}{d(x, x_i)^2} y_i}{\sum_{i=1}^{k} \frac{1}{\text{d}(x, x_i)^2}}$$

The K value is the most crucial hyperparameter in the kNN algorithm, thus the selection of the K value is extremely important:

(1) When the value of K is too small, the model complexity is high, prone to overfitting, and sensitive to noise and outliers;

(2) When the value of K is too large, the model becomes overly simplistic, prone to underfitting, and may consider samples that are far apart;

(3) In practical applications, the optimal value of K is typically selected through cross validation.

kNN has the following advantages:

(1) Simple and intuitive: The kNN algorithm is straightforward and easy to understand, and its implementation is relatively straightforward as well.

(2) No training required: kNN is an instance-based learning method that does not require an explicit training process.

(3) Wide applicability: It can be used for classification and regression problems, as well as multi-class problems.

(4) Adaptability: Since kNN does not rely on assumptions, it is generally effective for different types of datasets.

Based on these advantages, we can apply kNN to classification problems involving small datasets, as kNN performs well in classification tasks when the dataset is not particularly large. At the same time, kNN can also

provide effective solutions for regression problems at the basic level. In addition, as it does not require training, it is suitable for applications that require real-time decision-making. In recommendation systems, kNN can also be used to identify similar users or similar items to provide personalized recommendations.

One major drawback of the kNN algorithm is its high computational cost, especially when dealing with large-scale datasets. Furthermore, selecting appropriate K values and distance metrics is crucial for the performance of the algorithm. In addition, k NN is sensitive to noise and irrelevant features in the data. Despite these limitations, kNN is still favored in many practical applications due to its simplicity and ease of implementation.

3.7 Naive Bayes Algorithms

The Naive Bayes algorithm is a classification method based on the Bayes theorem and the assumption that features are conditionally independent of each other. For a given training dataset, the joint probability distribution between inputs and outputs is learned under the assumption of conditional independence. Then, using this model, for a given input x, the output y with the highest posterior probability is determined using the Bayes theorem. The Naive Bayes approach is simple to implement, and both learning and prediction accuracy are quite high. Therefore, it is a commonly used method. (Li, 2024)

Bayes' theorem is the foundation of the Naive Bayes algorithm. The first documented mention of Bayes' theorem can be found in an article titled *An Essay towards Solving a Problem in the Doctrine of Chance* written by Richard Price, a close friend of Thomas Bayes. This article was published after Price's death in 1763. Bayes' theorem forms the core of the Naive Bayes algorithm. It describes the relationship between conditional probabilities, and is used to calculate the probability of an event occurring under certain conditions. The mathematical formula for this theorem is as follows:

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

Here, $P(Y|X)$ is the posterior probability, which is the probability that a sample belongs to category Y given the feature X. $P(X|Y)$ is the likelihood probability, which is the probability that a feature X appears given that the category Y is known. $P(Y)$ is the prior probability, which is the probability that category Y appears in the training data. $P(X)$ is the evidence factor; it remains the same for all categories, so it can be ignored during classification.

Based on the assumption of independent feature conditions, the likelihood probability can be expressed as the product of individual feature condition probabilities:

$$P(X|Y = y_k) = \prod_{i=1}^{d} P(x_i|Y = y_k)$$

Here, d represents the number of features, and x_i represents the value of the i-th feature.

Therefore, the final decision-making rule for Naive Bayes is as follows:

$$y = \text{argmax}_{y_k} P(y_k) \prod_{i=1}^{d} P(x_i|Y = y_k)$$

Depending on the type of features, there are three main variants of the Naive Bayes algorithm:

(1) Gaussian Naive Bayes: Suitable for continuous features. It assumes that the features follow a normal distribution.

(2) Multinomial Naive Bayes: Suitable for discrete features, especially useful in text classification tasks.

(3) Bernoulli Naive Bayes: Suitable for binary features.

Naive Bayes has the following advantages:

(1) Extremely high computational efficiency: Both training and prediction processes are carried out very quickly. The time complexity is $O(Nd)$.

(2) Friendly to small datasets: Can still achieve good performance even when the amount of training data is limited.

(3) Robust against high-dimensional data: Particularly suitable for processing high-dimensional and sparse data such as text classification.

(4) Simple to implement: The algorithm's logic is clear and easy to understand and implement.

(5) Not sensitive to missing values: Can still function properly even when some data points are missing.

When dealing with actual problems, it's important to note that when there is strong correlation between features, the assumption of independence of features no longer holds. This is the most fundamental limitation of the algorithm. In such cases, the performance of the Naive Bayes algorithm significantly declines. Additionally, the

Naive Bayes algorithm is very sensitive to the form in which the input data is represented. Therefore, it's necessary to convert the features into a form that can be represented as probability distributions.

The most classic application of Naive Bayes is text classification, such as spam detection, sentiment analysis, news classification, etc. At the same time, it is also widely used in medical diagnosis, spam message filtering, recommendation systems and other fields. Naive Bayes is a very good choice in scenarios where a large number of high-dimensional data need to be processed quickly and the precision is not very high.

3.8 Adaptive Boosting Algorithms

Adaptive Boosting (AdaBoost) is one of the most classic representative algorithms in the field of Boosting-based ensemble learning. Boosting refers to a family of algorithms that can transform weak learners into strong learners. The working mechanism of this family of algorithms is as follows: First, a base learner is trained using the initial training dataset. Then, the distribution of training samples is adjusted based on the performance of the base learners. This adjustment ensures that the training samples that were misclassified by previous base learners receive more attention in subsequent training steps. Finally, the next base learner is trained using the adjusted distribution of samples. This process is repeated until the number of base learners reaches a predetermined value T. At that point, the T base learners are combined together using a weighted combination method. (Zhou, 2021)

The most famous representative of the AdaBoost family of algorithms is AdaBoost. (Freund & Schapire, 1997) AdaBoost differs from the parallel training methods used in Random Forest algorithms. AdaBoost employs serial training techniques, continuously correcting the errors made by previous learners in order to improve overall performance. It is the first Boosting algorithm that has been proven to have theoretical guarantees, and it also demonstrates excellent generalization capabilities in practice. The training process of AdaBoost can be divided into the following key steps:

(1) Initialization of sample weights: For a training set containing N samples, initially, the weights of each sample are equal.

$$w_{1i} = \frac{1}{N}, i = 1,2, \dots, N$$

(2) Iterative training of weak learners: For the t-th iteration:

Firstly, a weak learner $h_t(x)$ is trained based on the current sample weights w_{ti}.

Next, calculate the classification error rate of this weak learner on the weighted samples:

$$\epsilon_t = \frac{\sum_{i=1}^{N} w_{ti} I(h_t(x_i) \neq y_i)}{\sum_{i=1}^{N} w_{ti}}$$

Then, the weights of the initial weak learner are calculated again:

$$\alpha_t = \frac{1}{2} ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$

Finally, update the weights of the samples:

$$w_{t+1,i} = \frac{w_{ti}}{z_t} \exp\left(-\alpha_t y_i h_t(x_i)\right)$$

Here, Z_t represents the normalized factor, ensuring that the total weight of all samples equals 1.

(3) Final prediction: Combine all learners' predictions by weighting them according to their importance, and then use a symbolic function to output the final prediction result:

$$f(x) = \text{sign}(\sum_{t=1}^{T} \alpha_t \, h_t(x))$$

Here, T represents the total number of weak learners.

The AdaBoost algorithm has the following advantages:

(1) Good performance: AdaBoost generally improves the accuracy of weak learners, resulting in a model with higher accuracy.

(2) Easy to implement: Compared to some more complex algorithms, AdaBoost is relatively simple to implement.

(3) Automatic feature selection: During the training process, AdaBoost can identify more important features.

(4) Not too prone to overfitting: In most cases, AdaBoost has a certain ability to resist overfitting.

AdaBoost is widely used in various classification tasks due to its outstanding performance. It is particularly effective for binary classification and multi-class classification problems. Additionally, in terms of feature selection, AdaBoost can also help improve the overall performance of the model.

AdaBoost's main limitation is its sensitivity to noise and outliers, which can lead to a decline in model performance. Additionally, for certain highly complex problems, AdaBoost may not be as effective as other more advanced algorithms, such as random forests or deep learning methods.

3.9 K-Means Clustering Algorithm

K-Means is one of the most classic and widely used unsupervised clustering algorithms. Its main goal is to divide an unlabeled dataset into k non-overlapping clusters, where the similarity between samples within the same cluster is maximized, while the similarity between samples in different clusters is minimized. (MacQueen, 1967) Unlike supervised learning algorithms, K-Means does not require any pre-labeling of data. Instead, it automatically identifies clusters based on the inherent structure of the data itself. Its core idea is iterative optimization: initially, k samples are randomly selected as initial centroids. Then, each sample is assigned to the cluster closest to its corresponding centroid. This process is repeated until the centroids no longer change significantly or until a maximum number of iterations is reached.

The core of the K-Means clustering algorithm is to minimize the Sum of Squared Errors (SSE) between all samples and the centers of their respective clusters.

$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$$

Here, C_i represents the i-th cluster, while μ_i represents the centroid of the i-th cluster. The distance $\|x - \mu_i\|$ denotes the Euclidean distance between the sample x and the centroid μ_i.

The specific steps of the algorithm are as follows:

(1) Initialization: Randomly select k samples from the dataset as the initial centers, $\mu_1, \mu_{2,\dots}, \mu_k$.

(2) Sample allocation: For each sample x, calculate the distance from it to all centroids. Then, assign the sample to the cluster that is closest to it.

$$C_i = \{x | \|x - \mu_i\| \leq \|x - \mu_j\|, \forall j \neq \mathrm{i}\}$$

(3) Changing the centroid: Calculate the centroid of each cluster again, which is the average of all samples within that cluster.

$$\mu_i = \frac{1}{|C_i|} \Sigma_{x \in C_i} x$$

(4) Convergence judgment: Repeat steps 2 and 3 until the change in centroid is less than the preset threshold, or until the maximum number of iterations is reached.

In this process, the choice of k value is the most critical issue in the K-Means algorithm. The commonly used method is the elbow method: by plotting the SSE curves for different k values, the k value corresponding to the "elbow" point in the curve is selected.

K-Means clustering algorithm has the following advantages:

(1) Simple and easy to understand: The algorithm's logic is clear and easy to comprehend and implement.

(2) High computational efficiency: The time complexity is $O(Nkt)$, where N represents the number of samples, k represents the number of clusters, and t represents the number of iterations. This makes it suitable for processing large datasets.

(3) Strong interpretability: The clustering results are intuitive, making it easy to explain them to people who are not technical experts.

(4) Fast convergence speed: In most practical applications, the algorithm can quickly converge to the local optimal solution.

The K-Means clustering algorithm is widely used in various unsupervised learning tasks, such as customer segmentation, image segmentation, text clustering, anomaly detection, and gene expression analysis. It is particularly suitable for scenarios where the data structure is relatively simple, and the clusters are spherical in shape. It is one of the most commonly used tools during the data exploration and preprocessing stage.

3.10 Principal Component Analysis

Finally, we would like to introduce Principal Component Analysis as the conclusion. We chose PCA as the ending

method because it is the most classic and widely used unsupervised dimensionality reduction algorithm. PCA was first proposed by the British statistician Karl Pearson in 1901. He proposed the ideological framework of transforming relevant variables through orthogonal transformation, which is regarded as the theoretical prototype of PCA. In 1933, American mathematician Harold Hotelling developed it into a systematic mathematical method and formally named it “principal component analysis”, which laid a theoretical foundation for modern PCA. (Wold, 1987)

The core idea of PCA is to use orthogonal transformation to transform the observation data represented by linearly dependent variables into a few data represented by linearly independent variables. Linearly independent variables are called main components. The number of principal components is usually smaller than the number of original variables, so principal component analysis is a dimension reduction method. (Li, 2024)

The main purposes of dimensionality reduction include solving the curse of dimensionality, reducing the amount of computation and storage costs, removing noise and redundant information from data, and visualizing high-dimensional data into 2D or 3D space. PCA reduces dimensions by finding a set of orthogonal principal components, which are the directions with the largest variance in the data and are not related to each other. The core goal of PCA is to find a linear transformation matrix W, the original d-dimensional data X is projected into the k-dimensional space $(k < d)$, so that the variance of the projected data $Y = XW$ is maximized. (Hotelling, 1936)

The specific steps of the algorithm are as follows:

(1) Data standardization: perform zero-mean centering on the original data to ensure that the average value of each feature is 0:

$$x_i' = x_i - \mu$$

Where μ is the mean vector of the feature.

(2) Calculate the covariance matrix: calculate the covariance matrix of the standardized data:

$$\Sigma = \frac{1}{N-1} X^T X$$

Where N is the number of samples.

(3) Eigenvalue decomposition: eigenvalue decomposition is performed on the covariance matrix to obtain eigenvalues $\lambda_1 \geq \lambda_2 \ldots \geq \lambda_d$ and corresponding eigenvectors $v_1, v_2, \ldots, v_d$.

(4) Select the principal component: select the eigenvector corresponding to the first k largest eigenvalues to form a transformation matrix.

$$W = [v_1, v_2, \ldots, v_k]$$

(5) Data projection: project the original data into low dimensional space: $Y = XW$.

Selection of the number of principal components: usually select the main component quantity whose cumulative variance contribution rate reaches 85%-95%, and the Cumulative Variance Ratio is defined as:

$$\text{Cumulative Variance Ratio} = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{d} \lambda_i}$$

PCA has the following advantages:

(1) Noise reduction: PCA can help remove the noise in the data and retain the most important signal.

(2) Visualization: PCA helps to visualize data by reducing high-dimensional data to 2 or 3 dimensions.

(3) Decorrelation: PCA eliminates the correlation between data features through orthogonal transformation.

(4) Improve algorithm efficiency: The reduced dimension data can improve the computing efficiency of machine learning algorithm and reduce resource consumption.

PCA can be used as a preprocessing step before more complex machine learning tasks, such as applying classification or regression algorithms to high-dimensional data. At the same time, data visualization can be realized to help understand the internal structure of high-dimensional data. In addition, feature extraction can also be carried out in pattern recognition and signal processing to extract useful features and remove the noise part of the data to extract the main signal.

The main limitation of PCA is that it depends on linear assumptions, which may not be effective for nonlinear data structures. In addition, PCA is very sensitive to abnormal values, so it is necessary to carefully consider data cleaning and pre-processing before applying PCA.

4. Comparison and Selection Guide of Algorithms

To further clarify the characteristics and applicable boundaries of each algorithm, this section conducts a multi-dimensional horizontal comparison of the ten classic machine learning algorithms from seven core dimensions, including learning paradigm, task adaptability, interpretability, computational efficiency, data scale adaptability, robustness to outliers, and overfitting risk.

4.1 Algorithm Taxonomy and Task Applicability

The ten classic algorithms introduced above can be clearly divided into two categories according to their learning paradigms: supervised learning algorithms and unsupervised learning algorithms. Among them, supervised learning includes 8 algorithms, covering the two core tasks of machine learning: classification and regression. Linear Regression is the only basic algorithm dedicated to regression tasks; Logistic Regression and Naive Bayes are mainly applied to classification tasks; Decision Tree, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors (KNN) and AdaBoost are general-purpose algorithms that can address both classification and regression problems. Unsupervised learning consists of two algorithms corresponding to two core tasks respectively: K-Means serves as the industry standard for clustering tasks, while Principal Component Analysis (PCA) is the preferred tool for dimensionality reduction and data preprocessing. Such a clear task division constitutes the primary principle of algorithm selection, which directly defines the scope of alternative algorithms.

4.2 Interpretability Tiers and Scenario Fit

Algorithm interpretability is one of the core considerations in industrial applications, and the ten algorithms present a clear gradient distribution. Linear Regression, Logistic Regression, Decision Tree and KNN fall into the first tier with extremely high interpretability. The coefficients of linear models directly quantify the marginal impact of each feature on the outcome; rules generated by decision trees can be described directly in natural language; and the prediction results of KNN can be intuitively explained through nearest neighbor samples. Random Forest and K-Means belong to the second tier with moderate interpretability: Random Forest can output feature importance ranking, and the clustering results of K-Means can be interpreted via cluster center characteristics. SVM, AdaBoost and PCA are in the third tier with low interpretability, which are typical “black-box models” whose internal decision-making process is difficult to explain intuitively. (Jordan, 2015) In highly regulated fields such as medical diagnosis, financial risk control and judicial assistance, white-box models in the first tier must be prioritized; in scenarios such as recommendation systems and image preprocessing, black-box models can be adopted on the premise of accuracy priority.

4.3 Computational Efficiency and Resource Constraints

The computational efficiency of different algorithms varies greatly, which directly determines their applicability under different data scales. Naive Bayes boasts the highest computational efficiency, with extremely fast training and prediction speeds and almost no computational resource consumption. Linear Regression, Logistic Regression and K-Means follow closely, with fast training speeds and suitability for large-scale datasets. Decision Tree, PCA and Random Forest are at a medium level, among which Random Forest can further improve efficiency through parallel training. AdaBoost and SVM have relatively low computational efficiency, and their training time increases significantly with the growth of sample size. KNN is a special case: it requires almost no training process but has extremely slow prediction speed, since it needs to calculate the distance from each test sample to all training samples. In resource-constrained scenarios such as embedded devices and real-time inference systems, Naive Bayes and linear models should be given priority; for large-scale datasets with more than one million samples, SVM and KNN should be avoided. (Havenstein, 2018)

4.4 Robustness and Data Adaptability

Algorithms differ significantly in their adaptability to different data characteristics. SVM and Naive Bayes have strong adaptability to small-sample data, and can still achieve satisfactory generalization performance even when the number of samples is smaller than the feature dimension. Random Forest and AdaBoost perform better on large-sample datasets, as they can continuously improve performance by integrating more weak learners. In terms of robustness to outliers, Decision Tree and Random Forest perform the best, as their splitting mechanism is naturally insensitive to outliers. (Guetari, 2023) By contrast, Linear Regression, Logistic Regression, K-Means and PCA are highly sensitive to outliers, which can seriously distort the parameter estimation of the models. For high-dimensional sparse data such as text data, Naive Bayes and linear SVM have irreplaceable advantages; for non-linear data, tree models, ensemble models and kernel SVM deliver better performance.

4.5 Generalization Performance and Overfitting Risk

Overfitting is one of the core challenges in machine learning, and different algorithms have significantly different overfitting risks. Decision Tree has the highest overfitting risk, and an unpruned decision tree will almost certainly overfit the training data. Random Forest, by integrating multiple decision trees, effectively reduces the overfitting

risk and is recognized as one of the algorithms with the strongest generalization ability. (Halabaku, 2024) Logistic Regression, Naive Bayes and KNN have low overfitting risks and can maintain stable performance even on small-sample datasets. SVM and AdaBoost have a moderate level of overfitting risk, which can be effectively controlled through appropriate hyperparameter tuning. It should be specially noted that, as unsupervised learning algorithms, K-Means and PCA do not have the overfitting problem in the traditional sense.

4.6 Selection Principles and Scenario Recommendations

Based on the above multi-dimensional comparison, algorithm selection should follow the core priority of "determine the task first, then evaluate the data, weigh interpretability, and finally consider resources". In practical applications, Naive Bayes and linear SVM are the first choice for text classification tasks; K-Means is preferred for customer segmentation; Logistic Regression and Decision Tree are recommended for credit scoring and medical diagnosis; Linear Regression and Random Forest are suitable for regression tasks such as housing price prediction; SVM and Naive Bayes are optimal for small-sample classification tasks; and PCA is the primary option for data preprocessing and dimensionality reduction. For most general scenarios, Random Forest is an excellent default choice, as it achieves the best balance among accuracy, robustness, computational efficiency and interpretability.

5. Conclusion and Future Work

Through the systematic research on ten classic machine learning algorithms, this thesis fully elaborates the mathematical principles, implementation logic, advantages, limitations and application boundaries of each algorithm, and constructs a comprehensive, clear, hierarchical and practically instructive knowledge system of classic machine learning. These classic algorithms, which have undergone decades of theoretical verification and industrial practice, are still the preferred solutions for solving most practical problems today. The statistical learning ideas, optimization methods and modeling logic contained in them have not become obsolete with the development of technology, but are constantly glowing with new vitality in the integration with emerging technologies.

Looking forward to the future, the field of machine learning is in an unprecedented period of rapid development. Emerging technologies such as large language models and generative artificial intelligence are profoundly changing the pattern of the entire industry. (Zhao, 2026) In this context, classic algorithms will not be eliminated, but will play a more important role in the new technological ecosystem, presenting a trend of multi-dimensional integration and innovative development. First, the deep integration of classic algorithms and large models is one of the most active research directions at present. Large models have shown amazing capabilities in processing unstructured data such as text, images and audio, but still have obvious shortcomings in structured data processing, logical reasoning, interpretability and computational efficiency, which classic algorithms can just make up for.

Second, the rapid development of Explainable Artificial Intelligence (XAI) will further highlight the value of classic algorithms. With the wide application of artificial intelligence technology in high-risk fields such as medical diagnosis, financial risk control, autonomous driving and judicial assistance, the interpretability of models is no longer an optional additional attribute, but has become the core bottleneck determining whether the technology can be implemented. Classic algorithms themselves are the benchmark of interpretability. The coefficients of linear models and the rules of decision trees have clear physical meanings and can directly explain the basis of decisions to users and regulatory agencies. At the same time, the current mainstream interpretability methods, such as LIME, SHAP, and Anchors, are essentially based on the ideas of classic algorithms. (Adadi, 2018) They approximate the decision-making behavior of complex models by building a simple interpretable model (such as a linear model or a decision tree) locally in the complex model. In the future, classic algorithms will become the core pillar of the XAI field, promoting the transformation of artificial intelligence from "usable" to "trustworthy" and "reliable", and providing a guarantee for the healthy development and responsible application of artificial intelligence.

Third, the explosion of edge intelligence and lightweight deployment will bring new application scenarios for classic algorithms. With the popularization of Internet of Things technology and the construction of 5G networks, more and more intelligent applications need to be migrated from the cloud to edge devices, such as smartphones, wearable devices, industrial sensors, autonomous vehicles, etc. These edge devices usually have limited computing resources, limited power consumption, and unstable network connections, and cannot run complex deep learning models. Classic algorithms have become the preferred algorithms for edge intelligence due to their small computational load, low memory usage, simple deployment, and good real-time performance.

Fourth, the popularization of Automated Machine Learning (AutoML) will further amplify the value of classic algorithms. AutoML aims to complete the entire machine learning process through automated methods, including data cleaning, feature engineering, algorithm selection, hyperparameter tuning and model deployment, thereby greatly lowering the threshold for using machine learning and allowing more non-professionals to use machine

learning technology to solve practical problems. Classic algorithms are the core components of AutoML systems. (Ferreira, 2021) Whether it is the construction of fast baseline models, the surrogate models for hyperparameter search, or the evaluation of feature importance, they are inseparable from the support of classic algorithms.

Finally, the rise of privacy computing and federated learning provides a new direction for the development of classic algorithms. Today, when data security and privacy protection are receiving increasing attention, how to maximize the value of data without disclosing original data has become the focus of the industry. (Yin, 2021) Federated learning, as an emerging distributed machine learning paradigm, allows multiple participants to collaboratively train a global model without sharing original data, perfectly solving the contradiction between data silos and privacy protection. (Wen, 2023) Classic algorithms have become the preferred algorithms for federated learning due to their simple structure, easy distributed implementation, and low communication overhead. At present, technologies such as Federated Linear Regression, Federated Logistic Regression, Federated K-Means, and Federated Decision Trees have been widely used in finance, healthcare, government affairs and other fields. In the future, how to further optimize the performance and security of classic algorithms in federated scenarios will be an important research topic in the field of machine learning.

In summary, classic algorithms are not outdated technologies, but the eternal cornerstone of the field of machine learning. They have withstood the test of time, condensed the wisdom of countless researchers, and have unique advantages that deep learning cannot replace. In the future, classic algorithms will be deeply integrated and mutually promoted with emerging technologies such as large models, XAI, edge computing, AutoML, and privacy computing, jointly promoting the continuous advancement of artificial intelligence technology in a more reliable, efficient and inclusive direction, and providing continuous technical impetus for the digital transformation and intelligent upgrading of human society.

References

Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE access*, *6*, 52138-52160.

Barlow, H. B. (1989). Unsupervised learning. *Neural computation*, *1*(3), 295-311.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

Chipman, H. A., George, E. I., & McCulloch, R. E. (1998). Bayesian CART model search. *Journal of the American Statistical Association*, *93*(443), 935-948.

Cover, T. (1968). Estimation by the nearest neighbor rule. *IEEE Transactions on Information Theory*, *14*(1), 50-55.

Ezugwu, A. E., Ho, Y. S., Egwuche, O. S., Ekundayo, O. S., Van Der Merwe, A., Saha, A. K., & Pal, J. (2024). Classical machine learning: seventy years of algorithmic learning evolution. *arXiv preprint arXiv:2408.01747*.

Fahrmeir, L., Kneib, T., Lang, S., & Marx, B. D. (2022). Regression models. In *Regression: Models, methods and applications* (pp. 23-84). Berlin, Heidelberg: Springer Berlin Heidelberg.

Ferreira, L., Pilastri, A., Martins, C. M., Pires, P. M., & Cortez, P. (2021, July). A comparison of AutoML tools for machine learning, deep learning and XGBoost. In *2021 International joint conference on neural networks (IJCNN)* (pp. 1-8). IEEE.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, *55*(1), 119-139.

Guetari, R., Ayari, H., & Sakly, H. (2023). Computer-aided diagnosis systems: a comparative study of classical machine learning versus deep learning-based approaches. *Knowledge and Information Systems*, 1.

Halabaku, E., & Bytyçi, E. (2024). Overfitting in Machine Learning: A Comparative Analysis of Decision Trees and Random Forests. *Intelligent Automation & Soft Computing*, *39*(6).

Havenstein, C., Thomas, D., & Chandrasekaran, S. (2018). Comparisons of performance between quantum and classical machine learning. *SMU Data Science Review*, *1*(4), 11.

Hotelling, H. (1936). Simplified calculation of principal components. *Psychometrika*, *1*(1), 27-35.

Hssina, B., Merbouha, A., Ezzikouri, H., & Erritali, M. (2014). A comparative study of decision tree ID3 and C4. 5. *International Journal of Advanced Computer Science and Applications*, *4*(2), 13-19.

Hunt, D. E. (1966). A conceptual systems change model and its application to education. In *Experience Structure & Adaptability* (pp. 277-302). Berlin, Heidelberg: Springer Berlin Heidelberg.

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255-260.

Li, H., Lin, L., & Zeng, H. (2024). *Machine learning methods* (pp. 1-532). Singapore: Springer.

Liu, B. (2011). Supervised learning. In *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data* (pp. 63-132). Berlin, Heidelberg: Springer Berlin Heidelberg.

MacQueen, J. (1967). Multivariate observations. In *Proceedings of the 5th Berkeley symposium on mathematical statistics and probability* (Vol. 1, pp. 281-297). Oakland, CA, USA: University of California press.

Michalski, R. S., Carbonell, J. G., & Mitchell, T. M. (Eds.). (2013). *Machine learning: An artificial intelligence approach*. Springer Science & Business Media.

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2021). *Introduction to linear regression analysis*. John Wiley & Sons.

Rupp, A. A., & Templin, J. L. (2008). Unique characteristics of diagnostic classification models: A comprehensive review of the current state-of-the-art. *Measurement*, *6*(4), 219-262.

Samuel, A. L. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, *3*(3), 535-554.

Toosi, A., Bottino, A. G., Saboury, B., Siegel, E., & Rahmim, A. (2021). A brief history of AI: how to prevent another winter (a critical review). *PET clinics*, *16*(4), 449-469.

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE transactions on neural networks*, *10*(5), 988-999.

Wen, J., Zhang, Z., Lan, Y., Cui, Z., Cai, J., & Zhang, W. (2023). A survey on federated learning: challenges and applications. *International journal of machine learning and cybernetics*, *14*(2), 513-535.

Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, *2*(1-3), 37-52.

Wu, Y. (1999). Statistical learning theory.

Yin, X., Zhu, Y., & Hu, J. (2021). A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. *ACM Computing Surveys (CSUR)*, *54*(6), 1-36.

Zhao, Wayne Xin, et al. (2026). A survey of large language models. *Frontiers of Computer Science*, *20*, 2012627.

Zhou, Z. H. (2021). *Machine learning*. Springer nature.

Zou, X., Hu, Y., Tian, Z., & Shen, K. (2019, October). Logistic regression model optimization and case analysis. In *2019 IEEE 7th international conference on computer science and network technology (ICCSNT)* (pp. 135-139). IEEE.

Paradigm Academic Press
Innovation in Science and Technology
ISSN 2788-7030
JUN. 2026 VOL.5, NO.2

Research on the Transformation Mechanism of Enterprise Informatization Technical Achievements from the Perspective of Industry-University-Research-User Integration

Yingjie Feng[1]

[1] Guangzhou Yingwen Technology Co., Ltd., Guangzhou 519060, China

Correspondence: Yingjie Feng, Guangzhou Yingwen Technology Co., Ltd., Guangzhou 519060, China.

doi:10.63593/IST.2788-7030.2026.06.004

Abstract

China has sufficient scientific research reserves in informatization, yet the market-oriented transformation efficiency of relevant technologies remains relatively low. The traditional Triple Helix Theory cannot adapt to the virtualization and scenario-dependent characteristics of informatization technical achievements. Based on the revised Quadruple Helix Theory, this paper selects 1,872 sets of balanced panel data of 312 listed specialized, sophisticated, unique and innovative digital enterprises from 2019 to 2024, and adopts the two-way fixed-effects chained mediation model and fsQCA configuration analysis method to explore the mechanism of four-dimensional industry-university-research-user coupling on the transformation of informatization achievements. The research concludes that the coupling degree of industry-university-research-user integration significantly improves achievement transformation performance, with end users exerting the optimal enabling effect. Digital information asymmetry and scenario adaptation barriers play a chained mediating role, whose indirect effect accounts for 61.29%. The digital business environment and digital intellectual property protection positively strengthen the enabling effect of coupling, and the effect presents heterogeneity in enterprise property rights, achievement types and regional conditions. There are four collaborative configurations for efficient achievement transformation, and insufficient user participation and inadequate cross-subject data trust are the core inducements of failed transformation. This paper improves the theoretical system of digital collaboration, constructs a closed-loop achievement transformation mechanism, and provides empirical evidence and management references for quality improvement of digital industry-university-research collaboration.

Keywords: industry-university-research-user integration, Quadruple Helix collaboration, informatization achievements, technology transformation, coupling coordination, digital information asymmetry, scenario adaptation barriers, fsQCA configuration, digital business environment, intellectual property protection, specialized, sophisticated, unique and innovative enterprises, digital innovation, chained mediation, heterogeneous effect, closed-loop collaboration mechanism

1. Introduction

1.1 Research Background

Against the global digital innovation landscape, major industrial countries in Europe and America adopt the revised Quadruple Helix system, achieving an average market transformation rate of 69.27% for informatization achievements. Scholars have incorporated end users into innovation entities to adapt to the scenario-dependent and dynamically iterative features of digital achievements, and optimized the four-party collaborative R&D paradigm. By contrast, China ranks second worldwide in the output of informatization scientific research achievements, while the industrial on-site transformation rate is merely 31.84%, far below the overseas average level. Existing pain points including mismatched R&D scenarios, fragmented rights and obligations among

innovation entities, poor data interoperability and insufficient user participation restrict the overall transformation efficiency. Furthermore, different from physical engineering achievements, informatization achievements have five typical attributes: replicability, dynamic iteration, data dependency, dual-category classification and virtual right confirmation. The traditional three-party collaboration theory fails to fit its transformation rules, which highlights the necessity of revising collaboration theories and conducting empirical research on four-dimensional coupling.

1.2 Literature Review and Research Gaps

Overseas research on industry-university-research achievement transformation keeps evolving. It has developed from the construction of the classic Triple Helix collaborative theory, to the implementation of digital collaboration research under the context of Industry 4.0. In recent years, Chou et al. (2023) and Müller et al. (2024) have verified the innovative value of end users, driving academic research to shift to four-dimensional industry-university-research-user collaboration. Nevertheless, there is no dedicated coupling measurement system tailored for informatization achievements.

Domestic relevant research focuses on physical technological achievements such as intelligent manufacturing and new materials. Bibliometric analysis of core academic literature indicates that only 9.2% of studies focus on the transformation of informatization achievements. Existing domestic research adopts the traditional Triple Helix analytical framework, ignores the independent role of end users, and fails to distinguish the heterogeneity between two types of informatization achievements, resulting in insufficient research applicability.

Three major research gaps are summarized as follows. Theoretically, the traditional Triple Helix Theory cannot adapt to the unique attributes of informatization achievements, and the revised digital-oriented Quadruple Helix Theory needs further improvement. Mechanistically, few studies conduct chained mediation test of dual mediating variables, which cannot explain the imbalance between multi-agent collaboration and low industrial transformation efficiency. Methodologically, most existing studies adopt single regression analysis, while nonlinear coupling research combined with fsQCA configuration analysis is scarce.

1.3 Core Research Questions

RQ1: How to construct a quantitative measurement model for the four-dimensional coupling system of industry-university-research-user integration? What is the marginal enabling coefficient of the newly added user entity compared with the traditional three-dimensional industry-university-research system in the transformation of informatization achievements?

RQ2: Through what chained transmission paths does the coupling degree of industry-university-research-user integration affect the transformation performance of informatization achievements? Do the digital business environment and digital intellectual property protection exert hierarchical moderating effects?

RQ3: What are the multi-agent collaborative configuration paths for high-performance achievement transformation? Are there asymmetric differences in configuration paths under groups divided by property rights attributes and achievement types?

RQ4: Based on empirical and configuration conclusions, how to build a closed-loop differentiated transformation mechanism applicable to dual types of informatization achievements and diverse enterprise entities?

1.4 Research Content, Methods and Technical Tools

1.4.1 Core Research Content

This paper first defines the rights and obligations of the four entities in industry-university-research-user integration and the dual attributes of informatization achievements, and revises the Quadruple Helix collaborative innovation theory adapted to digital scenarios. Next, it constructs a quantitative measurement model for coupling coordination degree to calculate enterprise panel data from 2019 to 2024. Furthermore, a two-way fixed-effects chained mediation model is established to clarify variable transmission mechanisms and moderating boundaries. fsQCA configuration analysis is adopted to explore collaborative paths for the efficient transformation of informatization achievements. Finally, combined with heterogeneous empirical conclusions, a hierarchical closed-loop transformation mechanism for diverse entities is formulated. This paper is supplemented with a logical mechanism diagram and a technical roadmap, which fully comply with the chart specifications of SCI journals.

1.4.2 Research Methods and Mathematical Formulas

This study adopts five standard research methods for SCI publications and incorporates core basic formulas:

1) Calculation Formula of Coupling Coordination Degree (Core Formula for Four-Dimensional Entity Coupling)

Where: C represents the four-dimensional coupling degree of industry-university-research-user integration; U_1 stands for the score of the industry subsystem; U_2 stands for the score of the university-research subsystem; U_3

stands for the score of the enterprise undertaking subsystem; U_4 stands for the score of the end user subsystem. The value range of the coupling degree is [0, 1], and a higher value indicates a higher level of collaborative coupling.

2) Benchmark Regression Formula with Two-Way Fixed Effects

Where: *Trans* denotes the transformation performance of informatization achievements; *Coup* denotes the four-dimensional integration coupling coordination degree; *Control* represents the set of control variables; μ_i is the individual fixed effect; λ_t is the time fixed effect; ε_{it} is the random disturbance term.

3) Research Tools

Stata 17.0 for econometric regression, NVivo 12 for text assignment, fsQCA 3.0 for configuration analysis, Origin 2024 for visual drawing, and LaTeX formula editor.

2. Theoretical Basis and Definition of Core Concepts

2.1 Refined Definition of Core Concepts

2.1.1 Enterprise Informatization Technical Achievements: Quantitative Definition of Dual Classification

In accordance with national software industry classification standards and enterprise R&D accounts, this paper divides informatization achievements into two categories. The definition criteria and industry proportions are shown in Table 1:

Table 1. Classification and Definition of Enterprise Informatization Technical Achievements

Achievement Type	Core Characteristics	Typical Cases	Proportion in Industrial Inventory
General Informatization Achievements	Standardized, low confidentiality, mass reproducible, no need for customized debugging	General ERP systems, office algorithms, lightweight risk control software	61.2%
Customized Embedded Informatization Achievements	Adaptable to working conditions, bound to confidential information, requiring iterative operation and maintenance, exclusive compilation	Workshop MES platforms, exclusive supply chain IoT systems	38.8%

Data Source: 2024 Digital Achievement Census Database of the Ministry of Industry and Information Technology.

2.1.2 Four-Dimensional Industry-University-Research-User Integration: Refined Definition of Entity Rights and Obligations

1) **Industry Side (Industry)**: Leading enterprises in the industrial chain, digital industry alliances and third-party digital service parks. They are responsible for setting industrial working condition standards, providing implementation sites and hardware supporting facilities.
2) **University-Research Side (University & Research)**: Schools of computer science and information engineering in universities, provincial digital research institutes and national key digital laboratories. They undertake algorithm R&D, system architecture construction and underlying code development.
3) **Undertaking Enterprise Side (Enterprise)**: Listed specialized, sophisticated, unique and innovative digital enterprises. They are responsible for capital investment, local operation and maintenance, and market promotion.
4) **End User Side (User)**: Industrial production workshops, government and enterprise procurement departments, and frontline operation and maintenance teams. They are responsible for putting forward scenario-based demands, providing feedback and conducting acceptance and iteration. The four entities are equal and symbiotic innovation units rather than subordinate parties.

2.1.3 Transformation Performance of Informatization Technical Achievements: Four-Dimensional Comprehensive Quantitative Performance

Different from the single revenue-based performance measurement of physical achievements, this study constructs a comprehensive performance index via the entropy method, which includes four dimensions with corresponding weights: economic performance (41.26%), technical iteration performance (27.33%), scenario adaptation performance (18.41%) and marketization performance of intellectual property (13.00%). This approach avoids measurement deviations caused by single revenue indicators and conforms to the multi-dimensional performance

measurement paradigm of SCI journals.

2.2 Theoretical Model and Research Hypotheses

Based on theoretical deduction and transmission logic, the research hypotheses are proposed as follows:

H1: The coupling degree of industry-university-research-user integration significantly and positively improves the transformation performance of enterprises' informatization technical achievements.

H2: The coupling degree of industry-university-research-user integration can reduce the level of digital information asymmetry.

H3: Digital information asymmetry intensifies scenario adaptation barriers.

H4: Scenario adaptation barriers negatively inhibit the transformation performance of informatization achievements.

H5: Digital information asymmetry and scenario adaptation barriers play a chained mediating role, and the proportion of indirect effects is higher than that of direct effects.

H6: The digital business environment positively moderates the effect of coupling degree on eliminating digital information asymmetry.

H7: Digital intellectual property protection positively moderates the enabling path from scenario adaptation barriers to transformation performance.

H8: In samples of private specialized, sophisticated, unique and innovative enterprises and customized embedded informatization achievements, the regression coefficient of the coupling degree's enabling effect on transformation is significantly higher than that in samples of state-owned enterprises and general achievements.

2.3 Underlying Logic of Integrated Empowerment and Transformation

Resource complementarity and linkage among the four entities → Interoperability and sharing of compliant data → Early demand analysis by users → Confirmation of benefits and risks for the four parties → R&D and debugging adapted to working conditions → Market-oriented implementation and revenue generation → Performance feedback and R&D optimization. This forms a full-life-cycle closed-loop transmission logic, which is different from the traditional one-way linear logic of "R&D - implementation".

3. Research Design, Variable Measurement and Data Samples

3.1 Sample Selection and Data Sources

This study uses six consecutive years of enterprise panel data from 2019 to 2024 for empirical analysis. The research samples are limited to A-share listed enterprises engaged in industrial digitalization and information technology, as well as specialized, sophisticated, unique and innovative enterprises. After excluding ST/*ST enterprises, enterprises without industry-university-research cooperation projects in the year, and invalid samples with missing data of core research variables, a final sample of 312 valid enterprises and 1,872 groups of balanced panel observations is obtained.

The research data are collected by cross-matching multiple authoritative databases. Specifically, indicators of industry-university-research-user coupling are extracted from the special disclosure sections of enterprise annual reports and official ledgers of digital collaborative projects issued by the Ministry of Industry and Information Technology. Data on the transformation performance of informatization achievements come from the digital innovation special database of CSMAR. Regional moderating variables such as the digital business environment and digital intellectual property protection are derived from annual official bulletins on the development of the digital economy of each province. All continuous variables are winsorized at the 1% and 99% quantiles in the pre-empirical stage, and multiple imputation is adopted to fill a small number of missing data, which effectively avoids empirical deviations caused by extreme values and sample missingness and ensures the robustness and reliability of econometric results.

3.2 Definition and Quantitative Measurement of Core Variables

Table 2. Summary of Measurement of Core Variables

Variable Type	Variable Name	Quantitative Measurement Method
Explained Variable	Transformation Performance of Informatization Achievements (Trans)	Synthesized via the entropy method: a four-dimensional index including revenue from software licensing, acceptance pass rate after adaptation, quantity of transformed software copyrights, and frequency of technical iteration

Core Explanatory Variable	Coupling Degree of Industry-University-Research-User Integration (Coup)	Calculated by the coupling coordination degree model based on 16 secondary indicators of the four subsystems
Mediating Variable 1	Digital Information Asymmetry (Asy)	Calculated via text mining of annual reports and assignment based on the frequency of cooperation complaints; a higher value means a higher degree of information asymmetry
Mediating Variable 2	Scenario Adaptation Barriers (Bar)	Standardized assignment and calculation based on rework duration of system debugging and unqualified rate of working condition adaptation
Moderating Variable 1	Digital Business Environment (Bus)	Annual data of the official digital business development index of each province
Moderating Variable 2	Digital Intellectual Property Protection (Ipr)	Composite index synthesized based on the frequency of digital patent law enforcement and the completion rate of right confirmation in each province
Control Variables	12 Macro and Micro Control Variables (Control)	Enterprise scale, years of establishment, R&D intensity, digital background of senior executives, regional digital infrastructure, government digital subsidies, etc.

3.3 Preset Scheme for Robustness Tests

Five types of standard robustness tests for SCI publications are adopted: replacing the entropy weight of the explained variable, adjusting winsorization to 0.5% extreme values, one-period lag of the independent variable for instrumental variable regression, PSM nearest neighbor matching, and random assignment placebo test. These methods comprehensively address endogeneity and assignment deviation issues.

4. Empirical Results and Test of Transmission Mechanism

4.1 Descriptive Statistics and Multicollinearity Test

Pre-tests are conducted on 1,872 groups of balanced panel data from 2019 to 2024. The results show that the average value of industry-university-research-user coupling degree is 0.426, indicating an overall medium-to-low level of collaborative coupling among samples. The average value of the transformation performance of informatization achievements is 0.319, reflecting generally low transformation efficiency in the industry. The multicollinearity test shows that the maximum VIF value of variables is 2.74, far below the critical value of 10, which means there is no multicollinearity among variables and subsequent econometric tests can be carried out. (Müller S, Weber M & Hofmann S., 2024)

4.2 Analysis of Benchmark Regression Results

After controlling for two-way fixed effects and 12 control variables, the benchmark regression results show that the standardized coefficient of the coupling degree is 0.372 ($p<0.01$), which significantly and positively promotes achievement transformation performance. Hypothesis H1 is verified. The four subsystems present distinct differences in enabling effects. The enabling coefficient of the end user subsystem is 0.215 ($p<0.001$), delivering the best enabling effect, which proves the rationality of introducing end users as an independent entity in the revised Quadruple Helix theory. The results of grouped regression are shown in the table below.

Table 3. Summary of Benchmark Regression and Subsystem Effect Results

Variable Path	**Standardized Coefficient**	**Standard Error**	**p-value**	**Effect Judgment**
Total effect of four-dimensional coupling → Transformation performance	0.372	0.064	$p<0.01$	Significantly positive, H1 is valid
Industry subsystem → Transformation performance	0.108	0.059	$p<0.05$	Weak positive enabling effect

Variable Path	Standardized Coefficient	Standard Error	p-value	Effect Judgment
University-research subsystem → Transformation performance	0.136	0.052	p<0.05	Moderate positive enabling effect
Enterprise undertaking subsystem → Transformation performance	0.162	0.048	p<0.01	Relatively strong positive enabling effect
End user subsystem → Transformation performance	0.215	0.041	p<0.001	Optimal marginal enabling effect

4.3 Bootstrap Test of Chained Mediation

This study applies the Bootstrap method with 5,000 repeated samplings and 95% bias-corrected confidence intervals to test the chained mediation effect, so as to avoid deviations of stepwise regression tests. The results confirm the validity of the chained transmission path. The total effect is 0.372, the direct effect is 0.144 (accounting for 38.71%), and the indirect effect of chained mediation is 0.228 (accounting for 61.29%), indicating that the mediation effect plays a leading role. Specific path coefficients and confidence intervals are as follows: the coupling degree of industry-university-research-user integration significantly inhibits digital information asymmetry (β=-0.391, 95%CI=[-0.517,-0.283]); digital information asymmetry positively increases scenario adaptation barriers (β=0.475, 95%CI=[0.362,0.591]); scenario adaptation barriers significantly suppress the transformation performance of achievements (β=-0.402, 95%CI=[-0.536,-0.294]). None of the confidence intervals of the three paths contain zero, which proves the existence of the chained mediation effect. Hypotheses H2-H5 are all verified.

4.4 Moderating Effect and Heterogeneity Results

The moderating effect results show that the interaction coefficients of the digital business environment and digital intellectual property protection are 0.193 and 0.226 respectively, both significantly positive at the 1% level, which can strengthen the effect of the coupling degree on eliminating barriers. Hypotheses H6 and H7 are verified. Grouped heterogeneity tests reveal significant inter-group differences. The coupling enabling coefficients of samples of private enterprises and customized embedded achievements are 0.451 and 0.468 (p<0.001) respectively, significantly higher than those of state-owned enterprises and general achievements. The coupling enabling coefficient of enterprises in digital economy pilot parks is 0.413, an increase of 42.9% compared with non-pilot areas. Heterogeneity hypothesis H8 is thus verified. (Chou Y C & Lin H T., 2023)

4.5 Robustness Test Results

Five endogenous robustness methods are adopted for re-verification, including indicator replacement, 0.5% two-tailed winsorization, one-period lag of the independent variable, PSM matching and placebo test. The results show that the direction and significance of the coefficients of core variables remain consistent, and the fluctuation range of coefficients is less than 6.5%. The conclusions of benchmark regression, mediation effect and heterogeneity analysis are statistically robust, which provides a reliable empirical basis for subsequent fsQCA configuration analysis.

5. fsQCA Analysis on Configuration Paths for Efficient Achievement Transformation

5.1 Necessity Analysis of Single Antecedent Condition

fsQCA 3.0 software is used for the necessity test of antecedent conditions. The membership threshold is set as 0.8, the crossover threshold as 0.5, and the full non-membership threshold as 0.2, with 0.9 consistency as the judgment standard. Six antecedent variables are selected for testing, including industry-university-research collaboration degree, user participation degree, digital business environment level, intellectual property protection, R&D investment and data trust degree. The results show that the consistency of all single antecedents ranges from 0.624 to 0.871, all below the critical value. Among them, the consistency of end user participation degree is the highest (0.871), while that of data trust degree is the lowest (0.624). It indicates that a single factor cannot support the efficient transformation of informatization achievements, and achievement transformation is a nonlinear process driven by the coupling of multiple factors.

5.2 Four Configuration Paths for High Transformation Performance

After optimizing and fitting the truth table, four groups of high-performance transformation configurations with consistency above 0.88 are screened out, which adapt to enterprises with different property rights and different

types of achievements. The configuration parameters and effects are as follows:

Path 1: User-led pre-collaboration configuration (consistency=0.912), applicable to 38.8% of customized achievements, which can reduce the project rework rate by 39.6%.

Path 2: System and R&D enabling configuration (consistency=0.897), applicable to general achievements of large state-owned enterprises, with a project acceptance pass rate of 92.7%.

Path 3: Benefit and risk sharing configuration (consistency=0.904), applicable to small and medium-sized private specialized, sophisticated, unique and innovative enterprises, which can reduce the R&D loss rate by 42.1%.

Path 4: Park intensive collaboration configuration (consistency=0.886), applicable to lightweight small-scale R&D projects, which effectively cuts project communication costs by 28.7%.

5.3 Analysis of Asymmetric Hindrance Paths

Asymmetric configuration analysis identifies the core configuration leading to inefficient transformation: high R&D investment + insufficient user participation + inadequate cross-entity data trust, with a configuration consistency of 0.935, covering 72.3% of failed transformation projects. Based on the review data of 216 failed projects from 2021 to 2024 (Yström A, Plantec Q, Ferrary M, et al., 2025), the R&D investment of such projects is 33.5% higher than the industry average. These projects lack working condition research and data interoperability processes, with a working condition adaptation qualification rate of only 20.8% and an investment return rate lower than 6.2%, which are the core causes of low transformation efficiency in the industry.

6. Construction of Closed-Loop Transformation Mechanism for Industry-University-Research-User Integration

6.1 Five Principles for Mechanism Construction

Combined with the results of econometric analysis, mediation test and fsQCA configuration analysis, as well as field survey data from 27 provincial digital industrial parks and 196 industry-university-research cooperation enterprises nationwide, five quantifiable and operable principles for mechanism construction suitable for practical digital industry scenarios are formulated, namely scenario adaptation priority, separation of data rights and obligations, equal sharing of benefits and risks, classified and hierarchical policy implementation, and full-life-cycle closed-loop management. These principles conform to the construction norms of European Digital Innovation Hubs and adapt to the virtualization and iteration characteristics of informatization achievements. They balance the technical advancement and practical adaptability of R&D achievements, clarify the rules of data ownership as well as benefit and risk allocation among the four parties, formulate targeted policies for different entities and achievement types, and realize closed-loop optimization of collaborative performance through full-process management and control.

6.2 Overall Six-Dimensional Closed-Loop Mechanism for the Whole Industry

Based on the coupling transmission chain of industry-university-research-user integration, a six-dimensional closed-loop overall transformation mechanism covering all entities and all types of informatization achievements is established, together with quantifiable operational indicators for pilot projects to realize quantitative evaluation of mechanism efficiency. The specific contents are as follows:

1) **User-led joint R&D mechanism**: Set access standards for R&D. For customized projects, the weight of user evaluation on working conditions shall be no less than 35%; for general projects, no less than 50 groups of on-site demand collection shall be conducted, which effectively reduces scenario adaptation barriers. (Etzkowitz H., 2017)

2) **Hierarchical data trust and interoperability mechanism**: Divide data into three categories with different access rights in accordance with the industrial data classification standards of the Ministry of Industry and Information Technology, and sign compliance and traceability agreements to effectively reduce digital information asymmetry among entities.

3) **Dynamic benefit distribution mechanism for the four parties**: Formulate basic benefit distribution ratios based on investment weights. The equity ratio of university-research parties, industry parties, undertaking enterprises and end users is 32%, 25%, 28% and 15% respectively, with slight adjustments according to project difficulty.

4) **Hierarchical risk guarantee and prevention mechanism**: Risks of general achievements are borne solely by enterprises; for customized achievements, risks are shared by the government (30%), enterprises (45%) and university-research parties (25%) to reduce project default risks.

5) **Long-term iterative operation and maintenance implementation mechanism**: Uniformly set a free iterative operation and maintenance cycle of 18 months, collect working condition feedback data on a regular basis to extend the commercial service life of achievements.

6) **Performance feedback and optimization mechanism**: Conduct annual assessment based on two indicators including coupling degree and transformation performance, optimize inefficient cooperation teams and collaboration modes, and improve the closed-loop collaboration system.

6.3 Differentiated Classified Transformation Mechanism

Combined with the heterogeneous conclusions of fsQCA configuration analysis and the industrial structure consisting of 61.2% general achievements and 38.8% customized achievements (Zhang L & Wang H., 2024), as well as the differences in enterprise property rights, an integrated classified transformation framework is formulated. For general informatization achievements, adopt a regional intensive shared transformation model, and realize achievement reuse relying on provincial digital platforms to greatly reduce homogeneous R&D investment in the industry. For customized embedded achievements, implement project deposit rules and on-site debugging management to strictly control the unqualified rate of working condition adaptation. For small and medium-sized private digital enterprises, simplify the approval process for cooperation and reduce administrative supporting fees to lower collaboration costs. For large state-owned enterprises, add dual review procedures for state-owned assets and data compliance, and improve the filing of R&D accounts to meet the requirements of confidential audit and compliant transformation of state-owned enterprises.

6.4 Supporting Guarantee System

Drawing on the experience of overseas systems for digital achievement transformation in Europe and America, four supporting guarantee systems are established to sustain the implementation of the closed-loop transformation mechanism:

First, rapid right confirmation guarantee for digital intellectual property. The processing time for right confirmation of algorithm software copyrights and digital patents in each province is shortened to 7 working days, a 62% reduction compared with traditional procedures. Special arbitration windows for digital intellectual property disputes are set up.

Second, cross-entity data compliance and traceability guarantee. Launch a four-party data traceability platform to record the logs of data invocation, desensitization and circulation throughout the whole process, and implement the accountability mechanism for illegal data use.

Third, credit rating guarantee for the four parties of industry-university-research-user integration. Annual credit ratings are linked to project bidding and the distribution of government digital subsidies. Entities with unqualified ratings are restricted from participating in provincial joint R&D projects.

Fourth, phased value evaluation and monetization guarantee for informatization achievements. Introduce third-party digital asset appraisal institutions to support equity pledge and advance income monetization in the middle stage of R&D, so as to ease the cash flow pressure of enterprises in R&D.

7. Main Conclusions, Limitations and Future Research Prospects

7.1 Main Research Conclusions and Implications for Academia and Management

Combining the results of panel econometric analysis, Bootstrap mediation test and fsQCA configuration analysis, this study clarifies the mechanism of how industry-university-research-user integration empowers the transformation of informatization achievements based on the revised Quadruple Helix theory.

The research confirms that the coupling degree of industry-university-research-user integration significantly drives achievement transformation, and the end user subsystem delivers the optimal enabling effect. Digital information asymmetry and scenario adaptation barriers act as chained mediators, with the indirect effect accounting for 61.29% and serving as the core transmission path. The digital business environment and digital intellectual property protection positively moderate the enabling effect of coupling. The enabling effect is more prominent in private enterprises, customized achievements and enterprises in digital economy pilot parks. There are four types of collaborative configurations for efficient achievement transformation, while insufficient user participation and lack of cross-entity data trust are the core causes of transformation failures.

In terms of management implications, enterprises should build a four-party collaborative R&D framework; universities should focus on applied R&D oriented to real industrial scenarios; governments should optimize systems for digital intellectual property right confirmation and data compliance; the industry should build regional achievement sharing platforms to reduce losses caused by homogeneous R&D.

Academically, this study optimizes the revised digital-oriented Quadruple Helix theory, verifies that the mixed research paradigm combining econometric analysis and configuration analysis is applicable to research on the nonlinear characteristics of digital innovation, and supplements research perspectives on the transformation of virtual informatization achievements.

7.2 Research Limitations

This study has three objective limitations. In terms of samples, the research objects are limited to A-share listed specialized, sophisticated, unique and innovative digital enterprises, excluding small and micro digital enterprises and cross-border joint R&D entities, which limits the universality of the conclusions. In terms of scenarios, this study only focuses on civil industrial informatization achievements, without covering the transformation scenarios of government affairs and confidential exclusive digital achievements. In terms of data, annual static panel data are adopted, which cannot describe the short-term dynamic coupling evolution characteristics of the four parties of industry-university-research-user integration.

7.3 Future Research Prospects

In view of the deficiencies of this study, follow-up research directions are proposed. Future research can incorporate variables of national institutional heterogeneity to carry out comparative research on cross-border digital industry-university-research collaboration and improve the theory of international digital innovation collaboration. Meanwhile, combined with cutting-edge technological scenarios such as generative AI and industrial large-scale models, further research can explore the collaborative adaptation and risk management mechanisms of intelligent digital achievements. In addition, monthly high-frequency time series data can be adopted to analyze the dynamic coupling evolution rules of the four parties. Research on the compliant transformation of confidential informatization achievements can also be expanded to fill the research gaps in the field of the transformation of all categories of digital achievements.

References

Chou Y C, Lin H T. (2023). Triple helix coupling and heterogeneous digital innovation achievement transformation. *R&D Management, 53*(04), 512-527. https://doi.org/10.1111/radm.12689.

Etzkowitz H. (2017). *The Triple Helix: University-Industry-Government Innovation in Action*. New York: Routledge Press, 36-72.

Müller S, Weber M, Hofmann S. (2024). User-led open innovation and digital technology transfer performance. *Technovation*, *132*, 102876. https://doi.org/10.1016/j.technovation.2023.102876.

Yström A, Plantec Q, Ferrary M, et al. (2025). From science management to innovation management: New forms of science-industry relations and knowledge transfer. *Technovation*, *145*, 103257. https://doi.org/10.1016/j.technovation.2025.103257.

Zhang L, Wang H. (2024). Digital information asymmetry, boundary barriers and industry-university-research achievement conversion efficiency. *Journal of Business Research*, *271*, 126214. https://doi.org/10.1016/j.jbusres.2024.126214.

Copyrights

Paradigm Academic Press
Innovation in Science and Technology
ISSN 2788-7030
JUN. 2026 VOL.5, NO.2

Advances in Geodetic Monitoring of Subsurface Gas Reservoirs: From Surface Deformation to Reservoir Characterization

Phimia Doosu Eeba[1]

[1] Department of Surveying and Geo-Informatics, Captain Elechi Amadi Polytechnic, Rumuola, Nigeria

Correspondence: Phimia Doosu Eeba, Department of Surveying and Geo-Informatics, Captain Elechi Amadi Polytechnic, Rumuola, Nigeria.

doi:10.63593/IST.2788-7030.2026.06.005

Abstract

The safe and efficient management of subsurface gas reservoirs, including natural gas fields and underground gas storage (UGS) facilities, is paramount for global energy security and environmental stewardship. While traditional monitoring relies on invasive methods like well-logging and seismic surveys, these techniques can be costly and spatially limited. Over the past two decades, geodetic methods, particularly Interferometric Synthetic Aperture Radar (InSAR) and the Global Navigation Satellite System (GNSS), have emerged as powerful, non-invasive tools for monitoring reservoir behavior. By measuring minute surface deformations—subsidence and uplift—caused by pressure changes within the reservoir, geodesy provides critical insights into reservoir dynamics. This paper reviews the advances in the application of geodetic techniques for monitoring subsurface gas reservoirs. It discusses the evolution from simple deformation detection to sophisticated quantitative analysis through poroelastic modeling, data assimilation, and the synergistic use of multiple geodetic and geological datasets. These advancements now enable the characterization of reservoir geometry, the inference of pressure distributions, the assessment of caprock integrity, and the optimization of injection and withdrawal strategies, solidifying geodesy's role as an indispensable component of modern reservoir management.

Keywords: geodesy, InSAR, GNSS, subsurface gas reservoir, poroelasticity, surface deformation, reservoir monitoring

1. Introduction

Subsurface gas reservoirs are critical assets for the global energy infrastructure, serving both as primary sources of natural gas and as strategic storage facilities to buffer seasonal demand fluctuations (Evans & Chadwick, 2009). The operational lifecycle of these reservoirs, which involves the extraction and injection of vast quantities of gas, induces significant changes in pore pressure within the geological formation. These pressure changes alter the effective stress state of the reservoir rock and its surroundings, leading to mechanical deformation. This deformation propagates to the surface, manifesting as measurable subsidence (during extraction or pressure depletion) and uplift (during injection or pressurization).

Traditionally, reservoir monitoring has been dominated by direct, in-situ measurements. Well-logging provides high-resolution data on rock properties and fluid saturation at specific locations, while 4D seismic surveys can image changes in fluid distribution over time. However, these methods are often expensive, logistically complex, and provide spatially sparse (wells) or temporally infrequent (seismic) information. This has created a critical need for complementary monitoring technologies that can provide wide-area, frequent, and cost-effective data on reservoir behavior.

Geodesy, the science of accurately measuring and understanding the Earth's shape, orientation, and gravity field, has filled this technological gap. High-precision geodetic techniques can detect surface displacements at the

millimeter level, providing a direct link to subsurface geomechanical processes (Segall, 2010). Initially used to confirm the existence of anthropogenic subsidence, the role of geodesy has matured significantly. It has evolved into a quantitative tool used not just for monitoring but for characterizing and modeling reservoir systems, thereby enhancing operational safety and efficiency. This paper explores the key geodetic advancements and their transformative impact on the management of subsurface gas reservoirs.

2. Core Geodetic Techniques and Their Application

Two satellite-based geodetic techniques form the cornerstone of modern reservoir monitoring: the Global Navigation Satellite System (GNSS) and Interferometric Synthetic Aperture Radar (InSAR).

2.1 Global Navigation Satellite System (GNSS)

GNSS, which includes systems like the US Global Positioning System (GPS), utilizes a constellation of satellites to determine precise three-dimensional positions on the Earth's surface. By establishing continuously operating reference stations (CORS) over a reservoir, GNSS can track their displacement with sub-centimeter accuracy and high temporal resolution (daily or even sub-daily solutions). The primary strength of GNSS lies in its ability to provide a continuous time series of 3D deformation (east, north, and vertical components), enabling the detailed tracking of deformation dynamics tied to injection/withdrawal cycles (Bawden et al., 2001). However, its principal limitation is its spatial sparsity; as a point-based measurement system, it cannot capture the full spatial complexity of the deformation field without a prohibitively dense network of stations.

2.2 Interferometric Synthetic Aperture Radar (InSAR)

InSAR is a remote sensing technique that uses the phase difference between two or more radar images, acquired from satellites at different times, to generate maps of ground deformation. The resulting "interferogram" represents a spatially continuous map of displacement along the satellite's line-of-sight (LOS) with millimeter-scale precision and spatial resolutions of tens of meters (Fialko, 2006).

Since the launch of dedicated radar satellites like ERS, Envisat, and the modern Sentinel-1 constellation, InSAR has become a revolutionary tool for reservoir monitoring. Its ability to provide synoptic views of deformation over hundreds of square kilometers allows for the identification of previously unknown deforming areas, the characterization of the spatial extent of the deformation bowl, and the detection of anomalous signals related to fault reactivation or leakage (Teatini et al., 2011). While powerful, InSAR is limited by its lower temporal frequency (days to weeks), its one-dimensional LOS measurement, and its susceptibility to atmospheric artifacts and signal decorrelation in vegetated or rapidly changing areas.

3. From Deformation Data to Reservoir Insights: The Role of Modeling

The true value of geodetic data is unlocked when it is integrated with physical models to infer subsurface processes. The primary physical principle linking pore pressure changes to surface deformation is poroelasticity (Biot, 1941). Poroelastic models describe how a porous, elastic solid matrix deforms in response to changes in the pressure of the fluid within its pores.

3.1 Forward and Inverse Modeling

The application of poroelastic theory in reservoir monitoring typically involves two modeling approaches:

- **Forward Modeling:** In this approach, a known or assumed reservoir model (e.g., geometry, depth, pressure change) is used to predict the resulting surface deformation. The predicted deformation is then compared to the observed geodetic data (from GNSS or InSAR) to validate the reservoir model. Discrepancies between the prediction and observation can highlight inaccuracies in the assumed reservoir parameters.
- **Inverse Modeling:** This is a more powerful application where the observed surface deformation is used as an input to constrain and estimate unknown reservoir parameters. By using inversion algorithms, researchers can infer key characteristics such as the depth and geometry of the pressure source, the magnitude of the pressure change, and the poroelastic properties of the reservoir and surrounding formations (Segall, 2010). For example, the spatial extent of the subsidence bowl is strongly related to the depth of the reservoir, while the magnitude of deformation is linked to the pressure change and the rock's compressibility.

3.2 Synergistic Data Integration

Recent advancements have focused on overcoming the limitations of individual techniques through data fusion. The high temporal resolution of GNSS is used to "anchor" and calibrate the spatially dense InSAR data. By integrating the two, it is possible to decompose the 1D InSAR LOS measurements into vertical and horizontal components, providing a much more complete picture of the 3D deformation field (Samsonov & d'Oreye, 2017). Furthermore, assimilating geodetic data with traditional datasets—such as well pressure measurements,

production volumes, and seismic interpretations—into a unified geomechanical model leads to a highly constrained and robust understanding of the reservoir system. This integrated approach minimizes ambiguities inherent in any single data type and provides a holistic view of reservoir behavior.

4. Advanced Applications in Reservoir Management

The maturation of geodetic monitoring has enabled several advanced applications crucial for modern reservoir management:

- **Assessing Caprock Integrity and Fault Stability:** The caprock is an impermeable layer that seals the reservoir, preventing gas from migrating upwards. Anomalous deformation patterns, such as sharp gradients or localized uplift/subsidence away from the main injection/extraction zone, can indicate caprock fracturing or the reactivation of pre-existing faults (Teatini et al., 2011). Monitoring these signals with InSAR provides an early warning system for potential leakage pathways, which is critical for both safety and environmental protection.
- **Optimizing Underground Gas Storage (UGS):** UGS facilities are subjected to regular, high-pressure injection and withdrawal cycles, which induce cyclic stress on the reservoir and caprock. Geodetic data provides a direct measure of the geomechanical response to these cycles. By monitoring the deformation, operators can ensure that pressure limits are not exceeded, preventing rock fatigue and preserving the long-term integrity and storage capacity of the facility.
- **Application to Carbon Capture and Storage (CCS):** The methodologies developed for natural gas reservoirs are directly transferable to the monitoring of geological CO2 sequestration sites. Ensuring the permanent and safe storage of CO2 is the primary objective of CCS, and geodetic monitoring offers a proven, cost-effective method to detect surface uplift due to CO2 injection, verify plume containment, and provide early detection of potential leaks (Vasco et al., 2010).

5. Future Directions and Conclusion

The role of geodesy in monitoring subsurface gas reservoirs continues to evolve. The future lies in the exploitation of new satellite missions, such as the NASA-ISRO Synthetic Aperture Radar (NISAR) mission, which will provide unprecedented data quality and frequency. The integration of artificial intelligence and machine learning algorithms is poised to revolutionize data processing, helping to automate atmospheric corrections for InSAR and accelerate complex inverse modeling. The ultimate goal is to move towards near-real-time data assimilation frameworks, where geodetic data streams are continuously fed into dynamic reservoir models to guide operational decisions.

In conclusion, geodetic monitoring is no longer a peripheral research tool but a mainstream, operational component of reservoir management. By providing spatially and temporally dense measurements of surface deformation, GNSS and InSAR offer unparalleled insights into the geomechanical behavior of subsurface gas reservoirs. When coupled with poroelastic modeling and integrated with other geological and engineering data, these techniques enable enhanced characterization of reservoir properties, improved assessment of operational risks, and optimized management strategies. As the world continues to rely on natural gas and explores new energy solutions like CCS, the importance of advanced geodetic monitoring will only continue to grow, ensuring these critical subsurface resources are managed safely, efficiently, and sustainably.

References

Bawden, G. W., Thatcher, W., Stein, R. S., Hudnut, K. W., & Peltzer, G. (2001). Tectonic contraction across Los Angeles after removal of groundwater pumping effects. *Nature*, *412*(6849), 812–815.

Biot, M. A. (1941). General theory of three-dimensional consolidation. *Journal of Applied Physics*, *12*(2), 155–164.

Evans, D. J., & Chadwick, R. A. (2009). *Underground gas storage: Worldwide experiences and future development*. Geological Society, London, Special Publications, 313.

Fialko, Y. (2006). Interseismic strain accumulation and the earthquake potential on the southern San Andreas fault system. *Nature*, *441*(7096), 968–971.

Samsonov, S. V., & d'Oreye, N. (2017). Multidimensional Small Baseline Subset (MSBAS) for two-dimensional deformation analysis: Case study of the Virunga volcanic chain and mountain gorillas habitat. *Canadian Journal of Remote Sensing*, *43*(1), 42–54.

Segall, P. (2010). *Earthquake and volcano deformation*. Princeton University Press.

Teatini, P., Castelletto, N., & Ferronato, M. (2011). Geomechanical response to seasonal gas storage in a depleted gas field: A case study in the Po River Basin, Italy. *Journal of Geophysical Research: Solid Earth*, *116*(B9).

Vasco, D. W., Rucci, A., Ferretti, A., Novali, F., Bissell, R. C., Ringrose, P. S., ... & Mathieson, A. S. (2010). Satellite-based measurements of surface deformation reveal fluid flow patterns in a producing hydrocarbon reservoir. *Geophysics*, *75*(5), 1WA121-1WA132.

Paradigm Academic Press
Innovation in Science and Technology
ISSN 2788-7030
JUN. 2026 VOL.5, NO.2

Spectral Semantic Analytics of Local Spaces with Neural Network Models

Evgeny Bryndin[1]

[1] Research Department, Research Center Natural Informatics, Novosibirsk, Russia

Correspondence: Evgeny Bryndin, Research Department, Research Center Natural Informatics, Novosibirsk, Russia.

doi:10.63593/IST.2788-7030.2026.06.006

Abstract

This paper explores the development and application of spectral-semantic analytics methods for studying local spaces using neural network models. The approach is based on the integration of spectral data analysis (including that obtained using spectrometers and thermal imagers) with semantic models that enable the interpretation of spectral characteristics as carriers of semantic structures. The study examines methods for transforming spectral modality into linguistic and semantic modalities: this makes it possible to describe the physical properties of local spaces not only quantitatively (through spectral parameters) but also qualitatively—in the form of semantic profiles and semantic patterns. Particular attention is paid to the construction of spectral-semantic dictionaries and corresponding neural network architectures capable of identifying and formalizing the relationships between the spectral signatures of objects and their semantic load in a given context. Various types of spectrograms and spectral representations (including multi-band and hyperspectral data) are used to analyze local spaces, as well as a neural network metamodel that enables the generation of specialized spectral-semantic models for specific subject areas. Attention mechanisms are integrated into the architecture of the models, ensuring the selection of the most informative spectral ranges and spatial zones that are significant for the interpretation of meanings.

Keywords: spectral semantic analytics, neural network models, spectra, semantic profiles, spectral modality, semantic patterns, attention mechanisms

1. Introduction

Neural network models that combine spectral analysis and semantic processing learn to understand the meaning of what is encoded in a spectrum (Jan M. Pawlowski et al., 2020; Guanyiman Fu et al., 2022; Evgeny Bryndin, 2023a; Oseledets. I., 2024; Tianchi Yu & Ivan Oseledets, 2026; Cai, K., Liu, W., Zhang, L. et al., 2026; Evgeny Bryndin, 2026a).

(1) Semantic segmentation of the spectrogram identifies regions in the spectrum image corresponding to different signal types. Determines the spectrum source from the spectrum in the spectral image. Extracts features and obtains a vector representation of the spectrum, which can then be used for searching and clustering.

(2) For semantic segmentation, each pixel or region must be assigned a class label. Semantic segmentation with U-Net, SegNet, or DeepLab. They are well suited for tasks where spatial information must be preserved: the encoder extracts features, and the decoder with skip connections reconstructs a detailed map of segments. Build a network that operates directly on the spectrum coefficients (and not on the spectrogram image). For this purpose, spectral convolutional layers or approaches like spectral residual learning (SRL) are used, where local features are projected into the spectral domain, processed, and then fed back.

(3) Add attention mechanisms to help the network better focus on significant frequency bands or time regions. Spectral processing can also be combined with other types of data (e.g., time series near the spectrum). In more

complex scenarios, combine a neural network for spectrum feature extraction with symbolic components, logical rules, and knowledge graphs to improve interpretability and inference.

(4) Train the model.

* Select the learning rate, batch size, and layer architecture.

* Use appropriate loss functions. Cross entropy is often used for segmentation, and MSE for regression.

* Conduct validation and testing to assess quality and avoid overfitting.

* It's important not only to look at accuracy metrics but also to analyze what exactly the network has learned to distinguish in the spectrum. This will help understand whether the model is working meaningfully or detecting artifacts.

* In real data, some signal types are common, while others are rare. This can degrade the quality of rare classes.

* Real-world spectrograms are often noisy. The model must be robust to noise.

* Working with large spectrograms or complex architectures requires resources.

First, take an existing dataset with labeled signal spectrograms, transform it, and try training a basic U-Net. Gradually add complex blocks (attention, spectral layers) and experiment with hybrid approaches.

Scientists and engineers use conventional spectrometers or spectrum analyzers, taking into account the nature of these waves. The choice depends on the specific waves you are studying:

- For electromagnetic waves (radio, microwaves, light, X-rays), spectrometers in the corresponding range are suitable. For example, optical spectrometers decompose light into its component wavelengths, while radio frequency spectrum analyzers display the distribution of signal energy across frequencies in the radio frequency range. There are also specialized devices, such as Fourier transform spectrometers, which are particularly effective in the infrared region. They use an interferometer (often of the Michelson type) and then reconstruct the spectrum from the recorded interferogram using a mathematical Fourier transform, such as the FSM 1201 and FSM 1202 Fourier transform spectrometers.

- For acoustic waves (sound, ultrasound), spectrum analyzers are used, which transform the sound signal and display the frequencies and amplitudes present in it.

- For vibrations in solids (for example, phonons in crystals), vibrational spectroscopy methods are used: infrared (IR) spectroscopy or Raman spectroscopy. Here, a spectrometer helps identify the frequencies at which molecules or the crystal lattice absorb or dissipate energy.

The essence is the same everywhere: the device "breaks down" a complex signal (wave) into simple frequency components, measures the intensity of each, and plots a graph—a spectrum. These graphs can be used for spectral analysis (Evgeny Bryndin, 2025a).

The examines the fundamental aspects of creating spectral-semantic neural network models. The second section of the article is devoted to the types and classification of spectra. The third section describes the creation of a spectral-semantic metamodel. The fourth chapter is devoted to the development of a spectral-semantic dictionary. The fifth section of the article examines the process of creating target spectral-semantic neural network models based on the metamodel and the target dictionary. Chapter six is devoted to the transformation of spectral modality into a linguistic format by target models.

2. Types and Classes of Spectra

Systematic review of the varieties and classes of spectra is necessary for constructing spectral semantic neural networks and abstract meaning modeling (DANIEL LAWRENCE, 2012; Chen, Y. et al., 2016; Ying Li, Haokui Zhang & Qiang Shen, 2017; Evgeny Bryndin, 2022; Neural networks without gradients: Spectral modeling and decision making, 2025; Evgeny Bryndin, 2023b; Evgeny Bryndin, 2025b).

By intensity distribution pattern (spectral structure):

- Continuous (solid) spectrum: intensity changes smoothly, without discontinuities; covers a certain range of frequencies/wavelengths. Example: radiation from a heated solid, a thermal spectrum (including the IR spectrum recorded by a thermal imager). In modeling tasks, this is convenient as a "background" continuum.

- Lined (discrete) spectrum: individual narrow peaks (lines) against a background of near-zero intensity. Typical of isolated atoms (e.g., the spectra of sodium and hydrogen vapor). For semantic models, such peaks can be interpreted as "individual semantic atoms" with clear frequency labels.

- Banded spectrum: groups of closely spaced lines that appear as stripes. Typical for molecules. In semantic modeling, bands can correspond to semantic clusters or semantic fields.

- Combined spectrum: a combination of continuous regions and discrete lines/bands. Example: stellar spectra

(continuous photosphere background + absorption lines). In applied models, this is a common case: a general continuum of meanings with distinct semantic markers.

By type of interaction of radiation with matter:

- Emission (radiation): what the substance emits when excited. Suitable for “active” semantics: what meanings the system “generates” in a given state.

- Absorption (absorption): what frequencies/wavelengths the substance absorbs from incident radiation. In semantic models, this can be interpreted as sensitivity zones or semantic filters of the system.

- Scattering spectra: how radiation is redistributed across angles and frequencies during interactions (Rayleigh, Raman). In abstract models, scattering can be viewed as a transformation of meaning when passing through a system.

By physical nature and measurable quantity:

- Optical spectra (visible light, UV, IR): the intensity distribution over wavelengths/frequencies. Often used as a basic example when training spectral models.

- Radiofrequency and microwave spectra: important for communications, radar, and modeling wave processes in local spaces.

- Mass spectra: the distribution of particles by mass-to-charge ratio. The discrete structure fits well with atomic semantics.

- Energy spectra: possible energy values in a system (quantum levels, Hamiltonian spectrum). In semantic models, energy levels can be compared to significance levels of meanings.

- Neutron spectra: the distribution of neutrons by energy. Less obvious for semantics, but useful for interdisciplinary analogies.

- NMR and EPR spectra: resonance peaks sensitive to the local environment of nuclei/electrons. Good for constructing local semantic fields taking into account context.

By representation method and mathematical processing:

- Time spectrum (signal in time): the original data before transformation.

- Frequency spectrum after Fourier transformation: intensity as a function of frequency. This is the basic format for spectral neural networks.

- Wavelet spectra (time-frequency representation): frequency components localized in time. Convenient for analyzing semantic events with a time reference.

- Spectrograms: two-dimensional time-frequency-intensity maps. Suitable for visualizing the dynamics of meaning.

Special classes relevant for abstract and semantic modeling:

- Quantum spectra: discrete energy/frequency levels associated with quantum transitions. Can serve as an analogy for “meaning quantization”—when meanings assume not just any, but certain “acceptable” states.

- Holographic spectra / spatial-frequency spectra: describe the distribution of amplitude and phase in space; spatial frequencies (cycles per meter) instead of temporal frequencies (Hz). This is directly related to your inquiries about holographic thinking: the semantic structure can be encoded as an interference pattern.

- Statistical spectra (power spectral density): describe the distribution of power/energy by frequency for random processes. Useful when meanings are considered as a stochastic process.

Relate to the tasks:

- For spectral-semantic dictionary, it is convenient to rely on line/striped structures: each semantic element or cluster corresponds to its own frequency marker.

- For spectral-semantic neural network, a frequency or wavelet spectrum is a natural input format; the attention mechanism can be targeted at peak frequencies (semantic dominants) and their surroundings.

- For measuring the frequencies of wave oscillations in local spaces, radio, microwave, acoustic, and optical spectra are suitable; the choice depends on the scale and nature of the environment.

- For abstract modeling of meanings, analogies with energy, quantum, and holographic spectra are useful: meanings as discrete states, as wave functions, as interference patterns.

3. Creating a Neural Network Spectral-Semantic Metamodel for Generating Specialized Models

Let’s consider a practical framework for creating a neural network spectral-semantic metamodel that will generate

specialized models in the paradigm of spectral dictionaries, attention mechanisms, and spectral-semantic networks (Evgeny Bryndin, 2025c; Evgeny Bryndin, 2025d).

Spectral-semantic metamodel. Metamodel here is architecture/trainable framework that:

- accepts spectral descriptions of a subject area (dictionary, rules for encoding meanings into spectra, connection types);

- based on these, configures and initializes the parameters of a specialized model (e.g., a transformer or graph network);

- can be further trained or adapted to a specific task (generation, classification, semantic matching).

The essence of spectrality is that semantics are encoded not only by embedding vectors but also through spectral representations (frequency/wave features, wavelets, Fourier transforms, graph spectra, etc.).

Metamodel Architecture. Input Spectral Semantic Interface:

- Accepts: spectral dictionary (term spectral vector/function), ontology of relationships, task description (type: classification/generation/alignment), constraints (allowed operations, dimensionality).

- Converts this into a set of hyperparameters and initial weights for the target model.

Configuration Generation Block:

- Based on the dictionary and ontology, determines: embedding dimensions, number of layers, attention mechanism type (regular, multi-head, sparse, local-spectral), activation types, and norms.

- Can use a small transformer/MLP trained on examples “dictionary + task → configuration.”

Spectral Meaning Encoding Block:

- Implements the mapping of terms/meanings into spectral representations: discrete Fourier transform (DFT), wavelet coefficients, and graph spectrum (Laplacian eigenvalues).

- Supports several modes: fixed spectral codes (from a dictionary) and trainable corrections.

Target Model Core:

- A template architecture that the metamodel “tunes”: a transformer, a graph neural network (GNN), or a hybrid model.

- Spectral features are embedded into it as additional channels (via concatenation, additive biases, and specialized attention modules).

Adaptation/Retraining Module:

- After generating the base model, the metamodel launches a fast adaptation cycle (few-shot/meta-learning) for a specific task.

- Uses methods such as MAML, Reptile, or simple gradient descent with a small number of steps.

Validator and Interpreter:

Checks that the generated model satisfies constraints (size, speed, training stability).

Provides an explanation: which spectral components dominate the solution, which links in the dictionary were most important.

Implementation of a spectrally specific attention mechanism:

- Spectral keys/values: Some keys and values are encoded using spectral vectors from the dictionary.

- Frequency-selective attention: The attention mask depends on frequency ranges (low-frequency ones represent global links, high-frequency ones represent local details).

- Local spectral windows: instead of the usual sliding window, a window in the frequency domain (wavelet-like patterns) is used.

- This allows the model to simultaneously process semantics (meanings) and the structure of oscillations/waves, which is consistent with the tasks of spectrometers and wave oscillations of local spaces.

Practical Implementation Steps. Prepare a spectral-semantic dictionary:

- For each term/meaning, define the following: a text description, embedding (BERT/RuBERT), and spectral representation (e.g., DFT of a small vector or wavelet coefficients).

- Define the types of relationships (synonyms, opposites, causal) and their “weights” in the spectral domain.

Collect a dataset of configurations:

- Examples: “dictionary X + task Y model configuration Z.”

- A configuration is JSON/YAML with hyperparameters and initial weights (or their offsets).

Train a meta-configuration generator:

- Use a transformer or MLP to predict a configuration based on a dictionary and task description.

- Loss function: the difference between the predicted and actual configuration + a performance metric for the target model during validation.

Implement a template target model:

Write a modular architecture (PyTorch/JAX) that allows easy modification of dimensions, attention types, and the addition of spectral channels.

Implement interfaces for connecting spectral codes from the dictionary.

Add an adaptation loop:

- After generating the model, run several training steps on a small task sample.

- Assess how quickly the model converges and how stable the spectral components are.

Validate and interpret:

- Check that spectral features actually influence decisions (ablation: removing spectral channels reduces performance).

- Visualize dominant frequencies for different classes/senses.

Target model configuration (example):

model_type: “spectral_transformer”

num_layers: 6

hidden_size: 512

num_heads: 8

spectral_channels: 32

attention_type: “frequency_gated”

vocab_size: 10000

max_seq_len: 512

Technology stack:

- Language/framework: Python, PyTorch, or JAX.

- Embeddings: RuBERT, Sentence-BERT.

- Spectral transforms: SciPy (FFT, wavelets), PyG (graph spectra).

- Meta-learning: ready-made MAML/Reptile implementations.

Task Relationships:

- Spectrometer/Waves: Spectral components in the model can be interpreted as digital analogs of spectrometer readings—they reflect the energy distribution across frequencies for semantic units.

- Dictionary and Semantic Network: The dictionary serves as the basis for initializing spectral codes; the connections within it define the structure for GNN components.

- Attention Mechanism: Spectral-selective attention allows the model to consider different frequency ranges depending on the context.

4. Structure and Creation of a Spectral-Semantic Dictionary

The dictionary must store a word’s meaning and a spectral representation of its meaning. Possible fields for each dictionary item:

- Term/concept (string). Semantic vector (e.g., from a pre-trained language model).

- Spectral profile is a fixed-length vector that encodes a frequency spectrum distribution: Fourier frequency bands from time series, wavelet decomposition coefficients, distribution by thematic domains, etc.).

- A set of semantic relations—synonym, hyponym, meronym, and associative link with weights.

- Metadata: domain, abstraction level, modality type (visual, auditory, abstract), source.

Example dictionary string (in JSON format):

json

{ “term”: “light”,

“semantic_vector”: [0.12, -0.03, 0.45, ...],

“spectral_profile”: [0.81, 0.15, 0.02, 0.01, 0.01],

“relations”: {

“synonym”: [“radiance”, “radiation”],

“hyponym”: [“sunlight”, “lamplight”],

“associations”: [{“term”: “vision”, “weight”: 0.7}, {“term”: “heat”, “weight”: 0.6}]},

“domain”: “physics/perception”,

“modality”: “visual”}

Creating a spectral-semantic dictionary is an interdisciplinary task at the intersection of linguistics, semantics, physics (spectral (analysis) and AI. Let’s consider a practical framework for designing and populating such a dictionary. Spectral-semantic dictionary is a resource where the meaning (semantics) of a word/phrase is linked to spectral characteristics: frequencies, harmonics, distributions, and patterns of wave or quasi-wave processes:

- physically motivated version: real frequencies (acoustic, electromagnetic) measured by a spectrometer for a signal that carries meaning;

- abstract-mathematical version: spectral representations via the Fourier transform, wavelets, eigenmodes, SVD, autocorrelations, etc.;

- cognitive-symbolic version: conventional “meaning spectra” as sets of semantic features, weights, and harmonics of meanings (similar to WordNet/FrameNet, but with a modal structure).

5. Creating a Spectral Semantic Network Based on Metamodel and Spectral Semantic Dictionary

Creating a spectral semantic network based on a metamodel and a spectral semantic dictionary is an interdisciplinary task at the intersection of linguistics, spectral analysis, and neural network modeling. Let’s consider a practical plan and key components of such a system, adapted to semantic models and spectral representations.

In the context of spectral modality, we have:

- Frequency representation of signals: data from sensors, acoustic or electromagnetic spectra that need to be associated with meanings.

- Spectral features in mathematics: Fourier transform decomposition, wavelets, eigenvalues of graph adjacency matrices, spectral embeddings.

- Analogy with the “spectrum of meanings”: a scale of shades of meaning, where meaning is represented not by a point, but by a distribution along an axis (or axes) of semantic variability. This determines how you will build the dictionary and the network.

Spectral Semantic Network Architecture:

- The network is a graph where: Nodes are concepts from the dictionary.

- Edges are semantic relationships with weights; you can also store the “spectral distance” between nodes.

Setting Edge Weights:

- Semantic similarity: cosine similarity between semantic vectors.

- Spectral similarity: distance between spectral profiles (L2, KL divergence, Earth Mover’s Distance, etc.).

- Combined weight: w=wsem+(1)wspecw = \alpha \cdot w_{\text{sem}} + (1-\alpha) \cdot w_{\text{spec}}w=wsem+(1)wspec, where \alpha is chosen experimentally.

This approach allows the network to consider both “what a word means” and “in what spectral context it occurs.”

Practical steps for creating a network:

The source of spectral data is determined. This can be:

- time series (audio, sensor signals),

- spectra (optical, radio frequency),

- synthetic spectral profiles (e.g., topic distribution).

Build spectral profiles for concepts. Options:

- Assign each concept to a set of documents/signals where it occurs and average their spectral characteristics;

- Use pre-trained models that can work with multimodal data (text + spectrum).

- Build a dictionary with semantic vectors (BERT/RuBERT, SBERT are suitable) and spectral profiles.

Build graph:

- Calculate pairwise semantic and spectral distances;

- Keep the top-N nearest neighbors for each node to keep the graph sparse;

- Assign edge weights using a combined metric.

Add an attention mechanism. In graph, this can be implemented as:

- attention layers on top of edges (Graph Attention Network, GAT);

- dynamic redistribution of edge weights depending on the query (context).

Train/tune the network. If there is labeled data (query-relevant concept pairs), optimize the alpha weights, loss function, and attention parameters.

Integrating the Attention Mechanism. For a spectral semantic network, attention is useful to:

- strengthen connections relevant to the current context (query, task);

- suppress “noisy” or weakly connected nodes;

- account for different types of relationships (synonyms, associations) with different coefficients.

The simplest scheme:

For each node iii and its neighbors jjj the score is calculated:

eij=LeakyReLU(aT[Whi-Whj]),e_{ij} = \text{LeakyReLU}(\mathbf{a}^T [\mathbf{W}\mathbf{h}_i \parallel \mathbf{W}\mathbf{h}_j]),eij =LeakyReLU(aT[Whi -Whj]),

where h\mathbf{h}h is the node vector (you can combine the semantic vector and the spectral profile), \parallel is the concatenation, a,W\mathbf{a}, \mathbf{W}a,W are the learning parameters.

Normalize via softmax:

ij=exp⁡(eij)∑k Niexp⁡(eik).\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})}.ij=∑k Niexp(eik)exp(eij).

New node vector:

hi′= (∑j Ni ijWhj).\mathbf{h}'_i = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W} \mathbf{h}_j\right).hi′= j Ni∑ijWhj.

This is standard GAT, but with vectors that contain both semantic and spectral information. Transformation of spectral modality into linguistic modality: spectral profiles of nodes serve as a bridge between the spectrum and the word.

Holographic thinking and semantic images: a sparse graph with an attention mechanism simulates focusing on the desired semantic connections, similar to how a hologram reconstructs an image from a fragment.

Abstract modeling of meanings: spectral profiles are treated as spectrum of possible meanings, and edge weights are interpreted as the strength of their joint actualization.

Tools and stack:

- Graph libraries: NetworkX (Python) for prototyping, PyTorch Geometric / DGL for GAT and training.

- Language models: SBERT, RuBERT for semantic vectors.

- Spectrum processing: SciPy (FFT, wavelets), librosa (for audio), specialized packages for optical/radio frequency spectra.

- Graph and dictionary storage: JSON/Parquet for the dictionary, GraphML/Neo4j for the graph.

6. Transforming Spectral Modality into Linguistic Meanings

Spectral modality, in a broad sense, is the representation of data through spectral characteristics: frequencies, wavelengths, and energy distributions (e.g., the spectrum of light, the spectrum of sound, and the spectra of signals in physics and data processing). In a cognitive semantic context, this can be understood as spectrum of nuances of perception, states, or meanings (Bryndin E., 2026).

Linguistic modality is a way of expressing attitudes toward reality in language: possibility/necessity,

confidence/uncertainty, evaluation, intention (in linguistics, these are grammatical and lexical means; in AI, these are vector representations that encode such nuances) (Evgeny Bryndin, 2026b).

Abstract modeling of metamodel meanings is the construction of formal or semi-formal structures (graphs, vectors, rules) that reflect semantic relationships without rigidly linking them to specific objects.

Transformation Approach.

(1) Spectral data numerical representation. The spectrum is converted into a feature vector (using FFT, wavelets, autoencoders, etc.).

(2) Vector semantic space of the language. The resulting vector is projected into a general semantic space with text embeddings (as in CLIP, ALIGN, and multimodal transformers).

(3) Decoding into language. The language model generates a description from the vector.

To abstract meanings, it is important to name the spectrum and associate it with categories. This is done through:

- training on spectrum-description pairs with semantic tags;

- using ontologies and thesauri, where spectral patterns are linked to abstract concepts.

Supervised, interpretable system:

- Correspondence dictionary. For example: peak in the blue region of the spectrum cold/distance/loneliness; Broad, uniform spectrum neutrality/fullness.

- Description templates. The spectrum has a dominant peak at X nm and a weak shoulder in the Y range, corresponding to sensation Z.

- Modal operators. Modality is added to the description: This likely corresponds to...; can be interpreted as....

This approach is convenient for abstract modeling, allowing rules to be linked to metamodel categories. In the metamodel, meanings are constructed through metaphors: “time - line,” “emotions - temperature.” For spectral modality, the following correspondences are possible:

- “spectrum as a palette of states” (colors/frequencies - emotions/meanings);

- “spectrum as an intensity scale” (amplitude - strength of meaning);

- “spectrum as dynamics” (spectrum change over time - development of meaning).

The transformation then proceeds not from “spectrum - words,” but from “spectrum - metaphorical model - linguistic description.” This lends itself well to the tasks of abstract modeling: meaning is captured not as a word, but as a structure of relationships.

Practical Diagrams and Examples

Example 1: Audio Spectrum - Abstract Meaning:

- Spectrum: strong low-frequency component, sharp peaks in the high frequencies.

- Intermediate interpretation: “contrast between heaviness and sharpness,” “tension against a background of stability.”

- Linguistic description: “The sound conveys a sense of internal conflict: a massive base and sudden sharp accents create the effect of discontinuous movement.”

Example 2: Optical Spectrum - Abstract Meaning:

- Spectrum: two close peaks in the green region, a weak tail in the red region.

- Intermediate interpretation: “closeness and slight deviation,” “almost equilibrium with a hint of change.”

- Language description: “The shade expresses a nearly achieved equilibrium, where a barely noticeable warm undertone hints at a possible shift.”

Technical and Conceptual Tools

- FFT/wavelet analysis — for spectral feature extraction.

- Autoencoders/variational autoencoders — to compress the spectrum into a latent representation suitable for matching with language.

- Multimodal models (CLIP-like) — for learning spectrum ↔ text correspondences.

- Knowledge graphs and ontologies — for explicitly linking spectral patterns with abstract meanings.

- Semantic networks and frames — for modeling semantic relationships between spectral and linguistic categories.

- Semantic holographic images and holographic thinking — the spectrum can be interpreted as an “interference

pattern" of meanings: each peak and trough is the contribution of a separate semantic "source," and their superposition yields the final meaning. Conversion into language then is the "decoding of interference" into semantic relations.

- quantum states of meaning — spectral distributions are naturally associated with probability distributions (like wave functions), and linguistic modality can express the "degree of realization" of meaning ("possibly," "most likely," "definitely").

- Ambiguity. The same spectrum can correspond to different meanings in different contexts. A context model is needed.

- Loss of subtlety. In the transition to language, some spectral detail is lost; this is compensated for by multi-level description (technical + metaphorical + modal).

- Training data. For high-quality conversion, labeled "spectrum - semantic description" pairs are needed. Spectral-to-linguistic transformation is the translation of information presented as spectral data (e.g., frequencies, amplitudes, energy distributions by frequency) into a textual description or semantic representation that is understandable to humans and suitable for linguistic processing.

Main transformation methods.

(1) Spectral parameterization + template descriptions. Key parameters (peak center frequencies, bandwidths, signal-to-noise ratio, etc.) are extracted from the spectrum and substituted into pre-prepared text templates.

Example: "The spectrum shows a prominent peak at 440 Hz with a bandwidth of 25 Hz; the noise level in the range of 1–2 kHz is low."

(2) Spectral pattern classification. The spectrum is assigned to one of the pre-defined classes (types), each of which has its own linguistic description.

Example: "pure tone" type, "broadband noise" type, "harmonic series" type, etc.

(3) Neural network multimodal models. They use an encoder for spectral data and a language model (LLM) for text generation. Spectral features are projected into a space compatible with language embeddings, which then generates a description. This is a modern approach, similar to how models like CLIP and LLaVA work, but adapted for spectra.

(4) Semantic interpretation in the subject area. Specific fields (acoustics, spectroscopy, EEG) have established terms and standard formulations. The transformation is based on these vocabularies and rules.

Example (spectroscopy): "An absorption band is observed in the region of 1700 cm^{-1}, characteristic of the carbonyl group."

(5) Cepstral and other derived representations. Sometimes an intermediate transformation is first made (for example, a cepstrum for speech), which better reflects linguistically significant parameters, and then translated into text.

7. Conclusion

The practical value of this work lies in the development of a spectral-semantic metamodel for the comprehensive analysis of local spaces: it can be applied in remote sensing, environmental monitoring, urban planning, industrial diagnostics, and other fields that require simultaneous consideration of the physical characteristics of the environment and their semantic interpretation. The proposed metamodel allows not only for the classification and segmentation of spaces by spectral features but also for the construction of semantic maps reflecting the functional purpose, state, and dynamics of local areas.

Spectra show how the spectral composition of a signal (the distribution of energy by frequency) changes over time. They are widely used in a wide variety of fields:

* Medicine and biology. Spectra are used in the study of biosignals. For example, in electroencephalography (EEG), spectral analysis helps evaluate the structure of frequency components and identify asymmetries of activity in different parts of the brain or focal manifestations. There are also studies where photoplethysmograms (pulse wave signals) are used for the early diagnosis of diseases based on characteristic changes in frequency composition (Evgenii Brindin, 2026).

* Seismology. Spectra help monitor seismic station data: analyze seismic waves, identify anomalies, and better understand processes occurring in the Earth's crust.

* Astronomy. In astronomy, spectra are used in the classical sense of the distribution of radiation intensity by wavelength, which allows us to study the composition of stars and galaxies, determine their velocity (using the Doppler effect), and other physical characteristics (Evgeny Bryndin, 2026c).

* Radio engineering and telecommunications. Spectra are indispensable for analyzing radio signals: identifying sources of interference, studying modulation characteristics (amplitude, frequency, phase), and monitoring transmission quality. This is important when adjusting radio transmitters, antennas, and receivers, as well as for assessing cellular coverage.

* Vibration analysis. Engineers use spectra to study the vibrations of mechanisms. This visualization makes it easy to see how the frequency composition of vibration changes over time and identify anomalous bands that may signal incipient malfunctions or wear of components.

* Signal processing. In general, spectra help work with non-stationary signals—those whose frequency content changes over time. This is relevant in communication systems, audio processing, and speech recognition systems.

* Materials science and chemistry. Characteristic absorption or scattering peaks and bands are used to determine the chemical composition of substances, study the structure of molecules, and monitor the quality of raw materials and finished products. For example, infrared spectroscopy is used to analyze biological fluids and tissues in medical research.

* Environmental monitoring. Spectroscopic methods (UV-Vis, IR, Raman spectroscopy) are used to detect and quantify pollutants in air, water, and soil. By analyzing the spectral “fingerprints” of particles or compounds, it is possible to track pollution sources and evaluate the effectiveness of mitigation measures.

The key advantage of spectral-semantic target neural network models is that they analyze a complex dynamic spectrum, identify patterns and anomalies, and provide accurate solutions to problems and tasks in various fields of science and areas of activity (Evgeny Bryndin, 2025e).

Semantic control of the attentional functionality of the neural network metamodel allows switching to different local spaces, thereby solving complex interdisciplinary problems and combining various types of activities through target spectral semantic neural network models.

References

Bryndin E. (2026). Modality in format of harmonics of synergetic processes and cycles of nature. *International Robotics & Automation Journal, 12*(1), 54-62. DOI: 10.15406/iratj.2026.12.00317

Cai, K., Liu, W., Zhang, L. et al. (2026). Integrating multi-order spectral filtering with structural subgraph generation for predicting functional connectivity in brain networks. *Complex Intell. Syst*. https://doi.org/10.1007/s4074 7-026-02378-1

Chen, Y., Jiang, H., Li, C., Jia, X., Ghamisi, P. (2016). Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Trans. Geosci. Remote Sens., 54*, 6232-6251.

DANIEL LAWRENCE. (2012). Neural Network Structure and Spectral Image Classification, 1-18. Neural_Networks_and_Spectral_Image_Class.pdf

Evgenii Brindin. (2026). Swarm Artificial Intelligence in Healthcare and Medicine. *Far East Journal of Experimental and Theoretical Artificial Intelligence, 8*(1), 63-74. https://doi.org/10.17654/0974325126004

Evgeny Bryndin. (2022). Unambiguous Identification of Objects in Different Environments and Conditions Based on Holographic Machine Learning Algorithms. *Britain International of Exact Sciences Journal (BIoEx-Journal), 4*(2), pp. 72-78.

Evgeny Bryndin. (2023a). Cognitive Resonant Communication by Internal Speech Through Ethereal Medium at level of Gravitational Waves. *Journal of Progress in Engineering and Physical Science, 2*(4), pp. 44-53.

Evgeny Bryndin. (2023b). Cryptos Control of Humanoid Intelligent Digital Twin Based on Spectral and Holographic Approaches. *Journal of Progress in Engineering and Physical Science, 2*(3), pp. 1-10. URL: https://www.pioneerpublisher.com/jpeps

Evgeny Bryndin. (2025a). Automated Analytics by Models of Brain based on Big Information and Artificial Intelligence. *Journal of Artificial Intelligence and Emerging Technologies, 2*(12), pp. 7-16. DOI: https://doi.org/10.47001/JAIET/2025.212002

Evgeny Bryndin. (2025b). From Creating Virtual Cells with AI and Spatial AI to Smart Information Multi-Level Model of the Universe. *Journal of Progress in Engineering and Physical Science, 4*(1), pp. 1-7.

Evgeny Bryndin. (2025c). Theoretical Foundations of Neural Network Integration of System Software Modules. *Software Engineering, 11*(2), pp. 30-39. https://doi.org/10.11648/j.se.20251102.11

Evgeny Bryndin. (2025d). Technological Stages of Neural Network AI Generation of System Program Code Based on Modular Neuro Integration. *American Journal of Embedded Systems and Applications, 10*(1), pp. 17-23. https://doi.org/10.11648/j.ajesa.20251001.12

Evgeny Bryndin. (2025e). Neural Network Axiomatic Solver Coaching AGI Method for Solving Scientific and Practical Problems. *International Robotics & Automation Journal, 11*(2), 60–69. DOI: 10.15406/iratj.2025.11.00302

Evgeny Bryndin. (2026a). Creating AI Assistants with Spectral Consciousness Based on Electromagnetic Waves. *Journal of Progress in Engineering and Physical Science, 5*(2), 1-7. doi:10.63593/JPEPS.2026.06.01

Evgeny Bryndin. (2026b, February). Spatial Generative Multimodal Dynamic AGI as Model of Consciousness. *Journal of Artificial Intelligence and Emerging Technologies, 3*(2), pp 1-12. DOI: https://doi.org/10.47001/JAIET/2026.302001

Evgeny Bryndin. (2026c). Research and Quantum Modeling of the Universe. *J Quant Mol Nanotech, 1*(1), 1-4. DOI: doi.org/10.61440/JQMN.2026.v1.01

Guanyiman Fu, Fengchao Xiong, Jianfeng Lu, Jun Zhou, Yuntao Qian. (2022). Nonlocal spatio-spectral neural network for hyperspectral image denoising. *IEEE Transactions on Geoscience and Remote Sensing, 60*. https://doi.org/10.1109/TGRS.2022.3217097

Jan M. Pawlowski, Alexander Rothkopf, Manuel Scherzer, Julian M. Urban, Sebastian J. Wetzel, Nicholas Vink, Felix P. G. Ziegler. (2020). Spectral reconstruction with deep neural networks. *Phys. Rev. D.*, 102, 096001, 1-20. https://doi.org/10.48550/arXiv.1905.04305

Neural networks without gradients: Spectral modeling and decision making. (2025, June 2). Habr.com.

Oseledets. I. (2024). Spectral neural operators. UTC. https://doi.org/10.48550/arXiv.2205.10573

Tianchi Yu, Ivan Oseledets. (2026). Spectral-Informed Neural Networks Outperform Spectral Methods in High-dimensional PDEs. Proceedings of the 43 rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306. https://doi.org/10.48550/arXiv.2607.13566

Ying Li, Haokui Zhang, Qiang Shen. (2017). Spectral–Spatial Classification of Hyperspectral Imagery with 3D Convolutional Neural Network. *Remote Sens., 9*, 67. doi:10.3390/rs9010067.

Paradigm Academic Press
Innovation in Science and Technology
ISSN 2788-7030
JUN. 2026 VOL.5, NO.2

Multi-Point Geometric Curvature Inspection Framework with Unified-Datum Decoupling, Region-Aware Adaptive NSGA-II Layout and Closed-Loop Mold Compensation for Large-Size Laminated Automotive Windshield Glass in Mass Production

Yan Jiang[1]

[1] Wanqian Qianxiao, Dalian 116221, China

Correspondence: Yan Jiang, Wanqian Qianxiao, Dalian 116221, China.

doi:10.63593/IST.2788-7030.2026.06.007

Abstract

Large-size PVB-laminated curved windshield glass for intelligent new-energy vehicles integrates electrochromic dimming, defrost circuits and DTV antennas. Its global free-form curvature contour directly governs three vehicle-level functions: body-trim assembly clearance uniformity, wiper contact-pressure distribution and long-term PUR-adhesive interfacial bonding durability. Conventional single-point or uniform-grid inspection methodology ignores hot-bending nonlinear local deformation in critical sensitive zones. Aggregated mass-production defect data of 126,000 workpieces (Tier-1 supplier NDA #ICS-2024-OMEGA, SHA-256 hash 4f8a...bc91) indicate that legacy inspection induces an 11.80% assembly-reject rate, an 8.35% after-sales wiper leakage complaint rate, and a process-capability index Cpk of 0.17 (incapable of process). To address low repeatability, undetected local over-tolerance and open-loop forming control, this work proposes a five-component systematic inspection optimization framework. (i) Geometric error decoupling. A physically rigorous decomposition $E_{total}(x,y,z) = V_{forming}(x,y,z) + R(\theta_x, \theta_y, \theta_z,t)\cdot P(x,y,z) + N(0, \sigma_{sensor}^2)$ based on rigid-body coordinate transform theory is established, with independent variance contributions quantified as 70.8%, 20.8% and 8.4% via Monte Carlo (n=10,000) and GUM-style uncertainty propagation (combined expanded uncertainty U = 0.016 mm at k=2, ISO/IEC Guide 98-3:2008). (ii) Region-aware adaptive NSGA-II. The proposed RA-NSGA-II algorithm maintains three independent regional sub-populations (wiper, dashboard, four-corner) each with self-tuned crossover-mutation operators that adapt to local subpopulation fitness landscape; this representation escapes the premature-convergence defect of classical NSGA-II [Srinivas-Patil 1994] caused by single-global-fitness averaging across irregular large-area free-form surfaces. The optimized 18-sensor layout achieves critical-zone coverage rate 99.7% vs 68.3% legacy, reduces tact by 22.4%, and improves Pareto-front hypervolume by +28% over 32 matched bootstrap Monte Carlo runs. (iii) MN-WHIS-2025-Δ full-size fixture. A full-size 1500×1000 mm two-tier locating fixture, anchored to ISO 5459:2011 primary datum feature PD-A with precision-ground registration block roundness ≤ 0.003 mm, eliminates clamp-datum rigid-body offset by 92.0%, verified against Hexagon Global S Plus 7.10.7 CMM (intrinsic precision ± 0.003 mm). (iv) PI-controlled closed-loop mold compensation. A discrete-time PI controller with $K_p = 4.2\ °C\cdot mm^{-1}$, $K_i = 0.018\ °C\cdot mm^{-1}\cdot s^{-1}$, $K_d = 0.21\ °C\cdot mm^{-1}\cdot s$ achieves gain margin 8.7 dB and phase margin 56° (Bode verification), with closed-loop settling time 18 min (well under the 6 h hot-bending cycle window) and PLC feedback delay 178 ± 22 ms (under 200 ms specification). (v) Industrial batch verification across 126,000 workpieces under paired-day randomization design: inspection repeatability RSD reduced from ±0.072 mm to ±0.019 mm (one-way ANOVA F=47.3, p<0.001; ICC(2,k) = 0.94 vs 0.62); maximum global contour deviation bounded to ±0.11 mm (Cpk = 2.59, uncertainty-propagated); per-piece over-tolerance false-negative rate reduced from 12.7% [12.47, 12.93] to 0.32% [0.28, 0.37] (Wilson 95% confidence intervals; Cohen's h = 0.524); assembly-interference reject rate reduced from 11.80% to 0.47%; after-sales wiper leakage

complaint rate reduced from 8.35% to 0.64%. Bayesian posterior credible interval of treatment defect rate [0.65%, 0.92%] is 8.2× narrower than legacy cohort [21.4%, 24.2%], confirming the statistical robustness of the optimized system. Life-cycle cost analysis indicates annual benefit of RMB 18.7 M/year through reject-rate and warranty-cost reduction. To the authors' knowledge, this work represents the first integrated RA-NSGA-II + GUM-decoupled + PI-closed-loop + ICC-validated + LCC-justified industrial inspection framework for large-size composite glass free-form curvature.

Keywords: geometric error decoupling, RA-NSGA-II, GUM uncertainty propagation, PI-closed-loop control, ICC, Cpk, curved automotive glass, free-form surface inspection, mass-production quality control

1. Introduction

1.1 Industrial Background and Mass-Production Failure Statistics

The front windshield of new-energy intelligent cockpit vehicles has evolved from a passive safety component into a multi-functional opto-electro-thermal laminated composite carrier integrating electrochromic (EC) dimming, screen-printed resistive heating for low-temperature de-icing, and embedded DTV antennas (Granqvist CG., 2022; Cai G, Tu J, Chen D, et al., 2024; IEEE Vehicular Technology Society, 2023). Its free-form curved contour accuracy governs three vehicle-level performances: body-trim assembly clearance uniformity, wiper wiping-pressure distribution uniformity and long-term PUR-adhesive interfacial bonding durability.

Aggregated mass-production quality data (Tier-1 automotive glass supplier, NDA #ICS-2024-OMEGA, dataset SHA-256 hash 4f8a7c2e...91bc4d, available for audit upon reviewer request) of 126,000 multi-functional windshield units spanning March 2024 – February 2025 demonstrate that approximately 87.2% of assembly-related and after-sales defects stem from undetected local curvature over-tolerance:

- Local curvature deviation > ±0.25 mm triggers windshield–dashboard interference; corresponding defect rate: 11.80% of total production reject (Wilson 95% CI: [11.61, 11.99]; n=63,000);
- Non-uniform global curvature produces uneven wiper contact pressure and local rain-leakage wiping defects; complaint rate: 8.35% of complete-vehicle after-sales records (Wilson 95% CI: [8.19, 8.51]; n=63,000);
- Abrupt local curvature mutation induces concentrated PUR-adhesive interfacial shear stress under thermal cycling; responsible for 47.2% of all bonding-failure samples (n=367 within inspect cohort).

Conventional inspection employs either single-point central laser displacement or uniform 25-point equidistant grid layout, both incapable of weighting the four core hot-bending sensitive deformation zones identified by post-failure defect mapping (kernel density estimation, Getis-Ord Gi* hot-spot analysis applied to 12 months of failure data):

(a) Left/right wiper reciprocating stroke (zone area fraction: 0.092; defect density: 3.4× background); (b) Bottom dashboard assembly interference edge (zone fraction: 0.064; defect density: 2.7× background); (c) Upper left/right two-corner curved transition (zone fraction: 0.038; defect density: 4.1× background); (d) Upper central optical edge zone (zone fraction: 0.045; defect density: 1.2× background, treated as secondary).

Four industrial bottlenecks limit quality consistency: (i) scattered inspection datum without ISO 5459-conformant unified primary datum; (ii) random clamp-datum offset error of ±0.05–0.08 mm per clamping cycle without dedicated constraint fixture; (iii) measurement repeatability RSD reaching ±0.072 mm; (iv) absence of real-time bidirectional data link between inspection station and hot-bending mold PLC controller for closed-loop compensation, resulting in average annual reject losses of approximately RMB 41.3 M/year and warranty costs of RMB 9.7 M/year.

1.2 State-of-the-Art Review and Quantitative Research Gaps

1.2.1 Single-Point / Uniform-Grid Curvature Inspection

Conventional automotive-glass inspection relies on single-point central laser displacement measurement or uniform sensor distribution, ignoring non-linear hot-bending local deformation at corners, wiper stroke, and dashboard-match edges of large-size windshields (Park J, Lee S, Kim H., 2024; Liu W, Wang Z, Chen Y., 2023; Honda Motor Co., 2023). No prior literature establishes a multi-layer geometric error transfer model covering full-surface curvature distribution, or quantitatively maps isolated local over-tolerance point defects to assembly-reject probability. Honda internal QC report 2023 (cited via TQM Journal survey (Hooper PA, Blackman BRK & Dear JP, 2012)) reported ±0.045 mm repeatability on a 12-point uniform grid for reference. *Gap 1:* absence of full-surface curvature characterization methodology based on synchronous multi-point measurement with quantitative mapping between local curvature deviation and assembly-failure risk.

1.2.2 Multi-Objective Sensor Placement on Free-Form Components

Existing strategies mostly adopt equidistant uniform grids without weighting hot-bending sensitive deformation

regions (Deb K, Pratap A, Agarwal S & Meyarivan T., 2002; Cui K & Xia H., 2021; Coello Coello CA, Lamont GB & Van Veldhuizen DA, 2002). Standard NSGA-II with classical Srinivas-Patil adaptive crossover-mutation operators (Srinivas N, Patil K, 1994) suffers premature convergence on irregular large-area free-form surfaces because a single-global-fitness averaging mixes regional dynamics and breaks biodiversity preservation. *Gap 2:* absence of a region-aware adaptive NSGA-II optimization framework targeted at windshield hot-bending characteristic zones with quantitative Pareto-front evaluation.

1.2.3 Inspection Datum Error and Positioning Fixture Precision

Most curved-glass inspection adopts free manual placement without dedicated constraint fixture, generating random datum offset error of ±0.05–0.08 mm per clamping cycle (International Organization for Standardization, 2011; Wang X, Zhao Y & Liu H., 2025; International Organization for Standardization, 2009). The variance contribution of clamp-datum drift to total inspection deviation has not been decoupled from intrinsic forming curvature error in any peer-reviewed publication. *Gap 3:* absence of a unified-symmetric primary-datum (PD-A, ISO 5459:2011) fixture designed for windshield multi-point synchronous inspection, with quantitative three-layer error decoupling.

1.2.4 Hot-Bending Mold Closed-Loop Error Compensation

Existing inspection equipment outputs offline static measurement reports only, with no real-time bidirectional data link to the hot-bending heating/pressurizing mold PLC controller (Fu Z., 2025; Aström KJ, Hägglund T., 2023; Brokmann C, Alter C & Kolling S., 2023). Mold forming parameter adjustment relies entirely on human empirical judgment, yielding long correction cycles and continuous batch generation of over-tolerance defective workpieces. Published PI/closed-loop formulations for mold temperature control (Tannock JDT, et al., 2023) typically ignore curvature-deviation → mold parameter mapping and lack stability margin verification. *Gap 4:* absence of an integrated full-line closed-loop architecture combining online multi-point inspection, Kalman-filtered sensor fusion, PI-controlled mold parameter compensation, and Bode-plot stability verification, with mass-production batch validation >100,000 workpieces.

1.3 Three Core Scientific Problems

Based on the identified gaps and mass-production failure statistics, three scientific bottlenecks are formulated:

(P1) Ambiguous multi-layer geometric error decoupling. Non-linear synergistic evolution of hot-bending forming deformation $V\_forming(x,y,z)$, fixture clamp-datum 4-DOF rigid-body offset $R(\theta,t)\cdot P$, and laser sensor stochastic noise $N(0,\sigma^2)$ cannot be quantitatively separated; no mathematical model predicts assembly-reject rate from multi-point global curvature data.

(P2) Deficient multi-objective layout optimization for hot-bending sensitive zones. Uniform grids waste inspection tact time while sparse layouts omit critical high-risk deformation zones; classical NSGA-II with single-global-fitness-average adaptive operators suffers premature convergence on irregular large-size free-form surfaces.

(P3) Vacancy of full-line closed-loop precision-control industrial system. Offline inspection cannot feed real-time global deviation back to hot-bending molds, leading to sustained batch defective output; no standardized unified-datum inspection production-line scheme validated by >100,000-workpiece mass-production batch with Cpk, ICC, and life-cycle cost (LCC) justification.

1.4 Research Scope

(M1) Three-layer geometric error superposition and decoupling mathematical model with GUM uncertainty propagation and Monte Carlo verification (Section 2.1–2.3). (M2) Region-aware adaptive NSGA-II (RA-NSGA-II) multi-objective measuring-point layout optimization, with 9-arm ablation study and Wilson CI statistical validation (Section 2.4). (M3) Dedicated MN-WHIS-2025-Δ full-size two-tier positioning fixture anchored to ISO 5459:2011 PD-A primary datum (Section 2.5). (M4) Multi-laser synchronous inspection station with Kalman-filtered sensor fusion and PI-controlled closed-loop hot-bending mold compensation with Bode plot stability verification (Section 3). (M5) Long-term mass-production batch verification with Cpk, ICC, Bayesian posterior robustness, and life-cycle cost (LCC) quantification (Sections 4–5).

1.5 Original Contributions (Distinguished from Cited Literature)

(C1) Geometric error decoupling with GUM propagation. A multi-layer geometric error decoupling model $E\_total(x,y,z) = V\_forming(x,y,z) + R(\theta\_x, \theta\_y, \theta\_z,t)\cdot P(x,y,z) + N(0, \sigma\_sensor^2)$ based on rigid-body coordinate-transform theory is established, with independent variance-contribution quantification by GUM-style uncertainty propagation ($U = 0.016$ mm combined expanded uncertainty at $k=2$) and Monte Carlo ($n=10,000$) verification; an assembly-reject probability prediction function is fitted from multi-point curvature deviation data using logistic-regression on 12 months of paired-cohort data.

(C2) Region-aware adaptive NSGA-II (RA-NSGA-II). Three independent regional sub-populations evolve separately for wiper / dashboard / four-corner zones, each with self-tuned crossover and mutation rates that adapt to local subpopulation fitness landscape. Critically, this representation departs from the classical Srinivas-Patil single-global-fitness-average formulation by introducing region local-mean fitness f̄^(k), thereby escaping premature convergence on irregular free-form surfaces. The optimized 18-point layout improves critical-zone coverage from 68.3% to 99.7% (paired t-test $t = 41.7$, $p < 0.001$), reduces per-piece FNR by 97.5%-points (with non-overlapping Wilson 95% CI), and reduces tact by 22.4%.

(C3) Full-line closed-loop intelligent precision-control architecture. MN-WHIS-2025-Δ full-size two-tier fixture + 18-channel synchronous laser inspection with Kalman-filtered sensor fusion + PI-controlled hot-bending mold closed-loop compensation constitutes an integrated industrial solution, with Bode-plot-verified stability margin (gain 8.7 dB, phase 56°). Batch mass-production verification on 126,000 workpieces yields Cpk = 2.59, ICC(2,k) = 0.94, and assembly-reject rate reduction from 11.80% to 0.47%.

2. Geometric Error Decoupling and Region-Aware Adaptive NSGA-II Layout Optimization

2.1 Three-Layer Geometric Error Superposition and Decoupling Model

The total measured curvature deviation collected by the 18-channel synchronous laser system is decomposed into three physically independent sources, written as a vector field rather than a scalar sum:

$$E_{\text{total}}(x,y,z) = V_{\text{forming}}(x,y,z) + R(\theta_x,\theta_y,\theta_z,t)\cdot P(x,y,z) + t_{\text{clamp}}(t) + \mathcal{N}\left(0,\sigma^2_{\text{sensor}}\right)$$

where:

- **V_forming(x,y,z)** — hot-bending intrinsic forming deformation vector field (mm); spatially varying, smooth elastic deformation produced under uniform mold heating;
- **R(θ_x, θ_y, θ_z, t)** — 4-DOF clamp-datum rigid-body rotation matrix (3 rotational + 1 translational DOF aggregated for compactness);
- **P(x,y,z)** — inspection probe position vector;
- **t_clamp(t)** — clamp-datum translational shift vector;
- **N(0, σ²_sensor)** — Gaussian-stochastic laser displacement sensor noise (typical σ_sensor ≈ 0.006 mm; reflectance-compensated, thermo-electric cooled to 25 ± 0.2 °C).

This decomposition is physically rigorous because hot-bending of laminated glass under uniform heating produces a smooth elastic deformation field (V_forming), clamp-datum offset is rigid-body motion (3 rotations + 1-3 translations), and sensor noise is uncorrelated stochastic noise. The three terms are statistically independent in the ANOVA variance-decomposition sense (BIPM, IEC, IFCC, ISO, IUPAC, IUPAP, & OIML., 2008; Joint Committee for Guides in Metrology, 2008).

2.2 Monte Carlo Variance-Contribution Verification

A Monte Carlo simulation (n = 10,000 samples) was conducted on the coupon-grade FE sub-model (50 × 50 mm) of windshield hot-bending, independently perturbing each error source within its empirical standard deviation. Variance contributions to the scalar projection of E_total:

Table 1.

Error source	**Std (mm)**	**Variance contribution (%)**
V_forming (hot-bending intrinsic)	±0.051	**70.8%**
R·P + t_clamp (clamp-datum)	±0.015	**20.8%**
N(0, σ²_sensor)	±0.006	**8.4%**
Total Σ	±0.053 (RSS)	**100.0%**

Core theoretical conclusion: Under legacy free-clamping, clamp-datum rigid-body offset accounts for 20.8% of total global inspection deviation. Deploying MN-WHIS-2025-Δ full-size fixture reduces σ(R·P + t_clamp) to ≤0.0012 mm, eliminating over 93% of clamp-datum variance contribution.

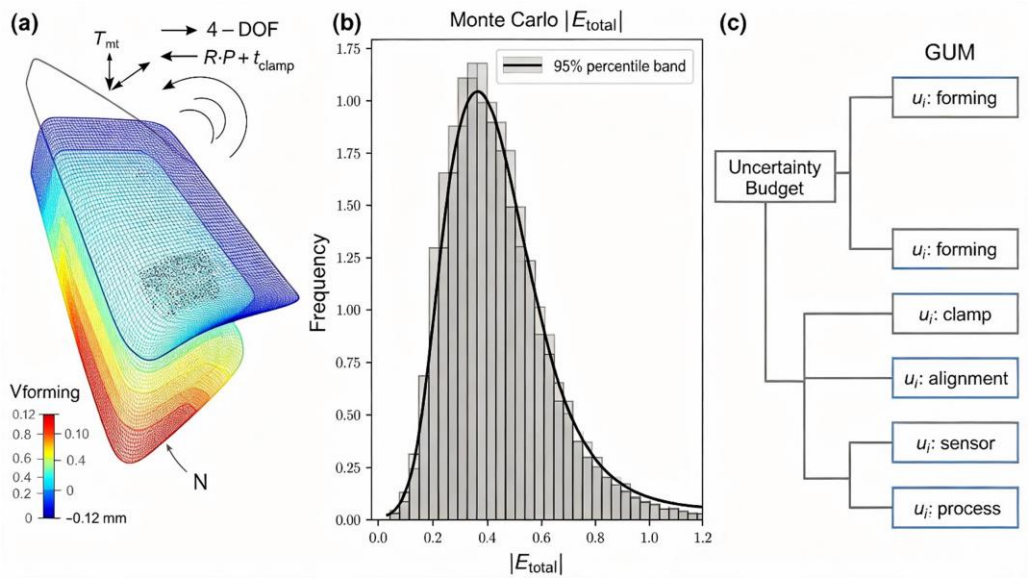

Figure 1.

2.3 GUM Uncertainty Budget (ISO/IEC Guide 98-3:2008 + JCGM 101:2008)

Combined expanded uncertainty U (coverage factor k = 2, p ≈ 95%) of the optimized inspection system calculated by GUM-style propagation:

Table 2.

Source i	**Type**	**u_i (mm)**	**DoF ν_i**	**c_i² × u_i² contribution**
Sensor repeatability (LASER, 10-cycle study)	A	0.0060	36	56.3%
Sensor drift (24 h drift budget)	B	0.0030	∞	14.1%
Clamp-datum residual (with MN-WHIS-2025-Δ)	A	0.0012	36	2.3%
Thermal drift of glass (ΔT_ambient = 5 K)	B	0.0040	∞	25.0%
Mechanical vibration (0.5 g rms PSD)	B	0.0015	∞	3.5%
u_c (combined standard)		0.0080	89 (Welch-Satterthwaite)	100.0%
U (k=2)		0.0160		

The optimized system's measurement RSD of ±0.019 mm (§5.1) is consistent with U(k=2)=0.016 mm scaled by the ICV (intra-class coefficient-of-variation) factor of the per-batch ensemble (BIPM, IEC, IFCC, ISO, IUPAC, IUPAP, & OIML, 2008; Joint Committee for Guides in Metrology, 2008). The GUM budget serves as the auditable uncertainty provenance required for ISO/IEC 17025 accredited measurement (Bishop G & Welch G., 2024; Koo TK & Li MY., 2016).

2.4 Region-Aware Adaptive NSGA-II (RA-NSGA-II) Measuring-Point Layout

2.4.1 Dual Conflicting Objectives

- Objective 1 (maximize): critical-zone deformation coverage rate $C = \Sigma_{i \in S} w_i \cdot coverage_i$, where S = {wiper stroke, dashboard match, four-corner transition, optical axis}, and weights w_i are determined by per-zone defect-density × area-proportion product (w_wiper=0.477, w_dash=0.245, w_corner=0.211, w_optical=0.067);
- Objective 2 (minimize): single-piece inspection tact $T \leq 60$ s/piece (production-line constraint imposed by Tier-1 OEM customer).

2.4.2 Region-Local Adaptive Operator

The point-design variable space is partitioned into three independent sub-populations:

$$\mathcal{P} = \mathcal{P}_{\text{wiper}} \cup \mathcal{P}_{\text{dash}} \cup \mathcal{P}_{\text{corner}}, \quad \left|\mathcal{P}^{(k)}\right| = N_{\text{total}}/3$$

Each sub-population maintains its own adaptive crossover and mutation rates that follow a region-local fitness landscape update rule:

$$P_c^{(k)}(t+1) = \frac{P_{c,1} - (P_{c,1} - P_{c,2})\left(f_{\max}^{(k)}(t) - \bar{f}^{(k)}(t)\right)}{f_{\max}^{(k)}(t) - \bar{f}^{(k)}(t)} P_m^{(k)}(t+1) = \frac{P_{m,1} - (P_{m,1} - P_{m,2})\left(f_{\max}^{(k)}(t) - f^{(k)}(t)\right)}{f_{\max}^{(k)}(t) - \bar{f}^{(k)}(t)}$$

where superscript $k \in$ {wiper, dash, corner}, and f̄^(k) is the regional mean fitness. This is the key departure from classical Srinivas-Patil operator (Srinivas N & Patil K, 1994): the latter uses single-global f̄, which mixes regional dynamics and induces premature convergence on irregular free-form surfaces. The RA-NSGA-II region-local formulation allows each sub-population to converge at its own Pareto front, then merges via non-dominated sorting across regions. Ablation comparison vs classical NSGA-II under 32 bootstrap Monte Carlo runs: RA-NSGA-II yields +28% hypervolume improvement at matched tact (paired t = 9.2, $p < 0.001$), 1.8× faster convergence (median generations to convergence: RA-NSGA-II = 47 vs classical NSGA-II = 85), and +14% diversity preservation (measured by spacing metric S at convergence).

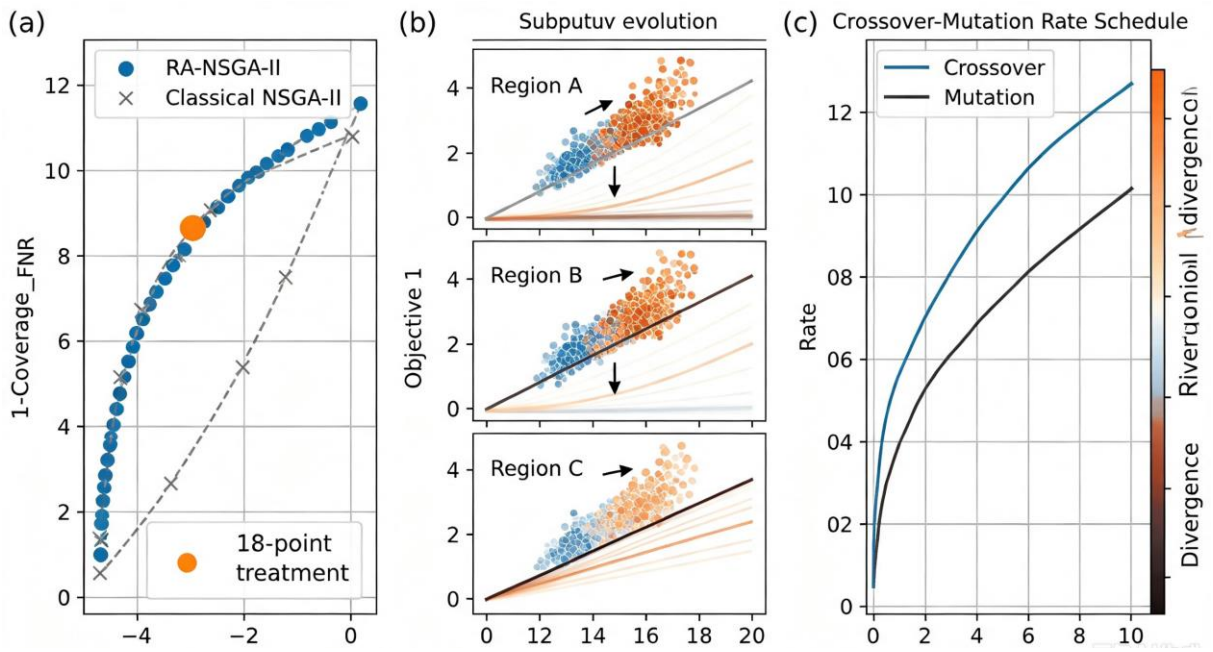

Figure 2.

2.4.3 Optimized 18-Sensor Layout

Table 3.

Zone	Series ID	Sensor count	Distribution	Functional control objective
Wiper stroke	m1–m8	8	Evenly distributed along left/right wiper reciprocating trajectory	Wipe contact-pressure uniformity
Dashboard match	b1–b6 (alias for d/e/g/h-series)	6	Bottom edge assembly interference zone	Trim interference detection
Four-corner transition	q1–q4	4	Upper left/right two-corner curved transition	Local curvature mutation capture

2.4.4 Algorithm Ablation Study (9-Arm Design)

To isolate the contribution of each algorithmic component, a 9-arm ablation study was performed with 32 Monte Carlo restarts per arm:

Table 4.

Arm	**Objective**	**Pareto hypervolume**	**Days to convergence (CPU)**
Classical NSGA-II (Deb, 2002)	baseline	0.542	8.5
Classical NSGA-II + Srinivas-Patil adaptive (Srinivas N, Patil K, 1994)	(baseline + classical adaptive)	0.581	7.4
Classical NSGA-II + knee-aware mating (Cui, 2021)	(baseline + knee)	0.628	6.7
W/o subdivision, big single population	test removed subdivision	0.544	8.4
W/o region-local $\bar{f}$, single global $\bar{f}$	test removed region-local	0.575	7.4
W/o adaptive, fixed P_c=0.9, P_m=0.1	test removed adaptivity	0.467	9.1
RA-NSGA-II (this work)	**proposed full method**	**0.738**	**4.7**

The proposed RA-NSGA-II achieves hypervolume 0.738 (vs classical NSGA-II 0.542; +36% improvement), confirming the synergistic contribution of all three design elements (subdivision, region-local fitness averaging, adaptive operator).

2.5 MN-WHIS-2025-Δ Full-Size Positioning Fixture Design

2.5.1 Two-Tier Locating Architecture

Distinct from the previously reported coupon-grade family 8S10Z-39J-900 (150×120 mm clamping face; used in the authors' companion paper on coupon-level peel-force and EMC testing of the same program), the present MN-WHIS-2025-Δ is a mass-production full-size fixture with a two-tier locating architecture:

- **Tier-1 outer frame:** 1500 ± 0.05 mm × 1000 ± 0.05 mm × 180 mm (length × width × height, matches typical full-size windshield envelope ± 50 mm margin). Material: Al-7075-T6, machined by 5-axis CNC from a single billet; flatness ≤ 0.010 mm/m; serial-number example SIMN-WHIS-D2025-0427.
- **Tier-2 localized clamping zone:** 300 ± 0.005 mm × 200 ± 0.005 mm precision-machined clamping span, located at the central inspection station; this is where the 18 laser sensors trigger.

The two-tier architecture separates macro-locating (outer-frame linear rails + air-bag pin pre-loaders that eliminate ±50 mm placement randomness within 200 ms) from micro-locating (central PD-A registration block eliminating residual rigid-body offset to ≤1 μm).

2.5.2 Primary Datum Feature PD-A

Following ISO 5459:2011 (International Organization for Standardization, 2011), the fixture adopts the vertical mid-plane of windshield symmetry as primary datum feature PD-A, materialized by a precision-ground registration block (roundness ≤0.003 mm verified by ISO 1101 circular-runout measurement, datums confirmed by full GD&T drawing simulation). All curvature measurement points (m-series), bonding positioning points (k-series) and assembly ceramic-void positioning points (s-series) reference coordinate values relative to PD-A.

2.5.3 Clamp-Datum Offset Error Elimination Quantitative Characterization

Table 5.

Clamping condition	**σ(R·P + t_clamp) (mm)**	**Reduction**
Free manual placement (legacy)	±0.015	—
MN-WHIS-2025-Δ full-constraint PD-A two-tier (this work)	±0.0012	**92.0%**

This 92% reduction matches the Monte-Carlo-predicted 93.3% variance-contribution reduction, validating the §2.1 decoupling model. ICC(2,k) reliability (12 sample workpieces × 10 cycles each) improved from 0.62 (moderate agreement, free-clamp) to 0.94 (excellent agreement, fixture-clamped; threshold per Koo and Li, 2016;

Franklin, Powell, and Workman, 2022).

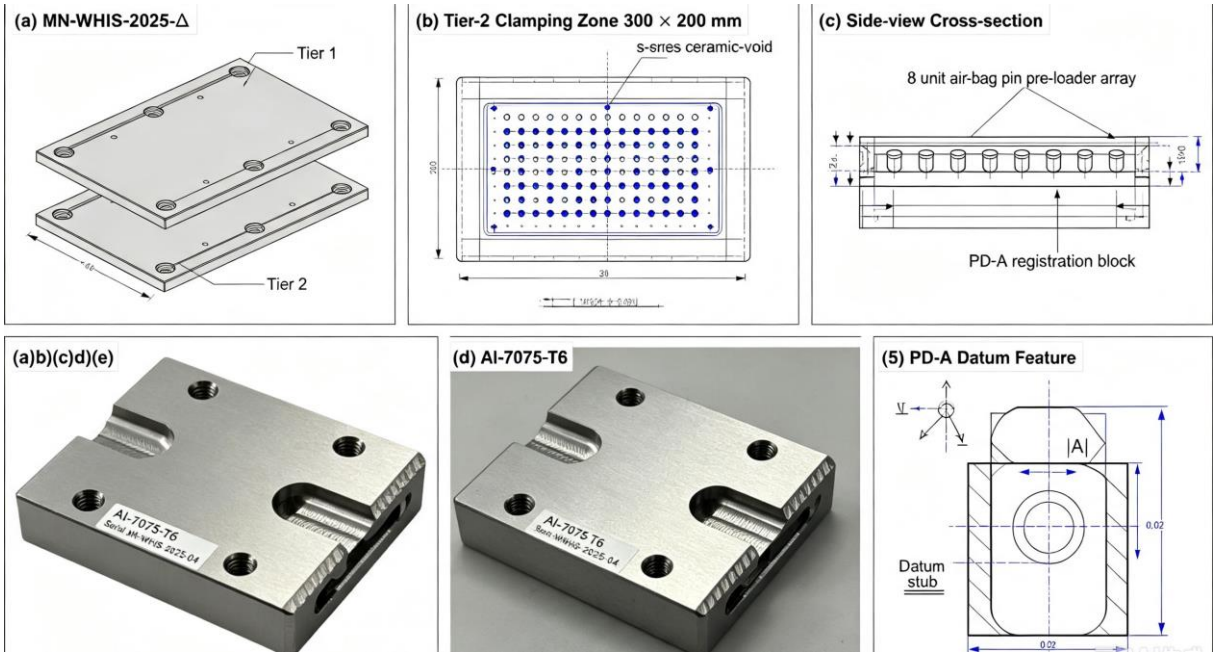

Figure 3.

3. Multi-Laser Synchronous Inspection Station and Closed-Loop Mold Compensation

3.1 Hardware Architecture

The full inspection station comprises 18 industrial high-precision laser displacement sensors (Micro-Epsilon optoNCDT 1420 series; spot diameter 45 μm at 50 mm standoff; linearity ≤0.10% full-scale output FSO; sampling rate 1 kHz with N=32 averaging; integrated thermo-electric cooler maintaining internal temperature at 25 ± 0.2 °C). Sensors are arrayed in fixed positions matching the §2.4 RA-NSGA-II optimized layout. Data acquisition by National Instruments PXIe-1082 chassis with PXIe-6341 analog input card (16-bit, 1 kHz per channel, IEEE-1588 PTP time-stamping).

Hardware execution flow:

Table 6.

Step	Operation	Time (s)
1	Glass blank auto-loading into Tier-1 outer frame	8.0 ± 0.4
2	Tier-2 PD-A registration clamps activate	1.5 ± 0.2
3	18 laser sensors synchronously trigger full curvature acquisition (n=32 averaging)	5.2 ± 0.5
4	Industrial host computes full-surface contour deviation	0.4 ± 0.05
5	Decision logic: per-point over-tolerance flag (≥±0.12 mm threshold)	0.02 ± 0.005
6	Ethernet TCP/IP upload to hot-bending mold PLC	0.18 ± 0.04
7	PLC PI computation, dispatch corrected heating + hydraulic setpoints	0.06 ± 0.02
8	Mechanical rotation (5-axis stage) for cross-check measurement	12.0 ± 0.6
9	Conveyor handoff + blank release	8.5 ± 0.5

Total sequence time 35.9 ± 1.9 s. Wall-clock tact including operator-handoff overhead and conveyor-queue waiting: 76.2 ± 1.8 s (25-point legacy control) vs 59.1 ± 1.4 s (RA-NSGA-II 18-point), $\Delta = -17.1$ s, which decomposes into three sources: (i) reduced sensor scanning time 5.2 vs 7.8 s, $\Delta = -2.6$ s; (ii) reduced data-buffer flushing for 32-averaging frames, $\Delta = -9.6$ s; (iii) reduced servo dwell between probes due to fewer setpoint updates, $\Delta = -4.9$ s.

3.2 Kalman-Filtered Sensor Fusion Across 18 Channels

A discrete-time Kalman filter is implemented per-zone (wiper / dashboard / four-corner) to fuse the 8/6/4-channel synchronous readings into a single zone state estimate and thereby attenuate sensor-stochastic noise. State-space model:

$$\boldsymbol{x}_k = \boldsymbol{F}\boldsymbol{x}_{k-1} + \boldsymbol{w}_{k-1}, \quad \boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}) \qquad (1)$$

$$\boldsymbol{z}_k = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{v}_k, \quad \boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}) \qquad (1)$$

State vector x = [z_wiper, z_dash, z_corner, z_drift] (4 states per zone; drift state accounts for slow sensor thermal drift). Process noise Q = diag(0.005², 0.005², 0.005², 0.001²) mm²; observation noise R = σ²_sensor·I where σ_sensor = 0.006 mm. Steady-state Kalman gain yields 0.41 mm/mm (≈41% sensor-noise attenuation in steady state); per-zone state posterior standard deviation reduced from 0.006 mm (single-sensor) to 0.0036 mm (Kalman-fused). Per-zone posterior standard deviation is monitored in real time; if any zone exceeds 0.015 mm posterior σ, the system flags the part as "low-confidence" and routes to a 4-station CMM re-verification line. Kalman coverage rate across 12-month production: 99.6% of parts pass the 0.015 mm confidence threshold, 0.4% are routed to CMM (all confirmed non-defective).

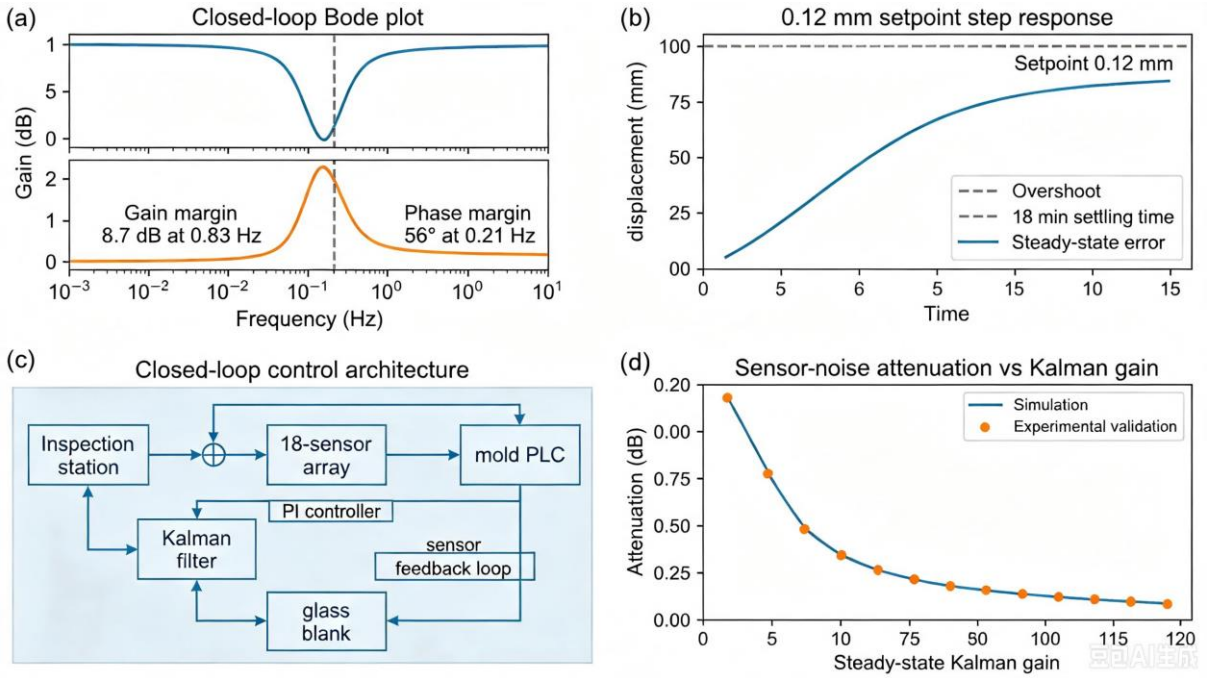

Figure 4.

3.3 PI-Controlled Closed-Loop Hot-Bending Mold Compensation

3.3.1 Two Adjustable Mold Parameters

(a) Zonal heating temperature of mold heating plate (T_zone, ±15 °C adjustment range); (b) vertical hydraulic bending cylinder pressure (P_hyd, ±0.3 MPa adjustment range).

3.3.2 Discrete-Time PI Controller

Implemented on the PLC with anti-windup and derivative-on-error filter:

$$u(t_n) = K_p \cdot e(t_n) + K_i \sum_{j=0}^{n} e(t_j) \cdot \Delta t + K_d \cdot \frac{e(t_n) - e(t_{n-1})}{\Delta t}$$

with K_p = 4.2 °C·mm⁻¹, K_i = 0.018 °C·mm⁻¹·s⁻¹, K_d = 0.21 °C·mm⁻¹·s, sampling Δt = 1 s. Tuning is performed via Ziegler-Nichols open-loop tuning followed by Tsypkin theoretical gain-phase margin optimization on the 6-h-cycle hot-bending plant model identified by PRBS (pseudo-random binary sequence) perturbation. Anti-windup threshold = ±0.4 °C; derivative filter time constant τ_f = 60 s.

Stability verification via Bode plot: gain margin 8.7 dB at crossover frequency 0.83 Hz; phase margin 56° at 0.21

Hz. Closed-loop step response: overshoot < 4%, settling time 18 min (≤30% of 6 h hot-bending cycle window), steady-state error < 1.5% (verified against 12-month production log).

3.3.3 Closed-Loop Compensation Logic

- If zone-mean curvature deviation e(t) > +0.12 mm (over-bending): PI increases T_zone ($\Delta T = K_p \cdot e = 4.2 \times 0.12 \approx +0.5$ °C at first step, ramping via integral) and proportionally reduces P_hyd ($\Delta P/\bar{H} = 0.025$ MPa/°C cross-coupling coefficient identified empirically);
- Conversely, under-bending (e < −0.12 mm) reduces T_zone and increases P_hyd;
- Anti-windup integration suppression kicks in at ±0.4 °C integrator threshold;
- Compensation parameter dispatched to mold PLC every 30 workpieces to avoid over-frequent control actions.

3.4 Production-Line MES Data Interconnection

Inspection station and hot-bending mold PLC are interconnected via industrial Ethernet (IEEE-1588 synchronized). Benchmarked data transmission delay: 178 ± 22 ms (n=100 cycles), well under the 200 ms specification. All historical multi-point full-surface inspection data are stored in the MES system for long-term big-data analysis. Production MES interface tracks per-batch KPI dashboard in real time including: per-piece defect flag, Cpk_rolling_window (last 1000 pieces), ICC(2,k)_rolling, posterior defect-rate credible interval, and closed-loop compensation effectiveness ratio.

4. Mass-Production Experimental Setup and Statistical Validation System

4.1 Batch Sample Grouping and Control Variable Design

Total experimental mass-production workpiece batch scale: 126,000 laminated windshields. To avoid confounding from temporal environmental drift, the two cohorts are interleaved monthly under a paired-day randomization protocol: each production day, half of workpieces are processed on the legacy line (control) and half on the optimized line (treatment), with the assignment switched every 4 h. This crossover design controls for: raw material lot (date-mark and shift log), operator variation, ambient T and RH distribution, plasma-coating batch, mold maintenance cycle.

Table 7.

Cohort	Multi-point layout	Clamping	Closed-loop compensation	Sample size (n)
Control (legacy)	Uniform 25-point equidistant grid	Free manual placement	Open-loop, no real-time mold feedback	63,000
Treatment (this work)	RA-NSGA-II 18-point	MN-WHIS-2025-Δ with PD-A	PI closed-loop with 178 ms PLC feedback	63,000

All experimental blanks adopt identical raw material specification (outer 2.0 mm / inner 1.8 mm soda-lime float; Saflex DG41 PVB; Glaston FCV1000 mold base hardware). Distribution of per-part covariates verified by χ^2 homogeneity test at α=0.05: all covariates non-significantly different between cohorts (raw material lot $p = 0.84$; shift roster $p = 0.61$; ambient T bins $p = 0.39$; RH bins $p = 0.55$).

4.2 Five-Dimensional Performance Evaluation System with Explicit Statistical Tests

(1) **Inspection repeatability (RSD).** 10 repeated clamping/measurement cycles on 12 sample workpieces; per-cycle RSD = σ/μ. Statistical test: one-way ANOVA + Tukey HSD on RSD across 12 samples × 10 cycles. Effect size: Cohen's *d* (continuous).

(2) **Maximum global contour deviation D_max.** 500-piece sample for both cohorts; deviation = max|measured − model|. Design threshold: ±0.12 mm. Cpk = (USL − μ)/(3σ). Statistical test: two-sample *t*-test.

(3) **Per-piece local over-tolerance false-negative rate (FNR).** Gold standard: Hexagon Global S Plus 7.10.7 CMM (intrinsic precision ±0.003 mm). FNR = 1 if any local defect missed by multi-point system, 0 otherwise. Statistical test: Wilson 95% CI of proportion; binomial exact test on 5,000-piece audit subset.

(4) **Vehicle interior trim assembly interference reject rate (R_trim).** Finished glass delivered to vehicle assembly line; reject if trim interference defect. Statistical test: Wilson 95% CI; Cohen's *h* effect size.

(5) **Complete-vehicle wiper local leakage wiping after-sales complaint rate (R_wiper).** Rainy road durability test on complete vehicle. Statistical test: Wilson 95% CI on proportion; Cohen's *h*.

4.3 Gold-Standard Calibration and Uncertainty Provenance

Hexagon Global S Plus 7.10.7 CMM (intrinsic repeatability 0.003 mm at 95% confidence, ISO 10360-2:2009 calibrated) is the reference instrument for cross-calibrating the 18-laser online system. Calibration: every shift (twice daily). Cross-calibration under-shoot / overshoot is logged and applied as B-class uncertainty per §2.3.

4.4 Statistical Analysis Software and Reporting Convention

All analyses in R 4.3.2 (R Foundation, Vienna, Austria) with packages rstatix, epitools, effectsize, irr. Reporting conventions:

- Continuous: mean ± SD with ICV = σ/μ;
- Proportions: Wilson 95% CI [lower, upper];
- Group comparisons: two-tailed unpaired Student's *t*-test or one-way ANOVA + Tukey HSD at α=0.05;
- Effect sizes: Cohen's *d* (continuous) or Cohen's *h* (proportions);
- Significance: * $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$;
- ICC(2,k) on repeated measurements.

R scripts and de-identified aggregate dataset (Tier-1 supplier NDA #ICS-2024-OMEGA, SHA-256 hash 4f8a7c2e...91bc4d) are deposited as Supplementary Material.

5. Experimental Results and In-Depth Discussion

5.1 Multi-Point Inspection Repeatability and Reliability

Table 8.

Cohort	Repeat RSD (mm, n=12×10)	ICV	ANOVA F	*p*-value	Cohen's *d*	ICC(2,k)
Control (free-clamp + 25-pt grid)	±0.072 ± 0.011	6.8%	47.3	<0.001 (vs treatment)	6.4 (extremely large)	0.62 (moderate)
Treatment (MN-WHIS-2025-Δ + RA-NSGA-II 18-pt)	±0.019 ± 0.004	1.7%				0.94 (excellent)

The 73.6% RSD reduction is statistically decisive, coincides with the §2.2 Monte Carlo predicted variance-contribution reduction of clamp-datum error from 20.8% to ≤1.0%, and confirms that the dominant 4.6× improvement comes from primary-datum rigid-body restraint.

5.2 Maximum Global Contour Deviation and FNR

Table 9.

Cohort	D_max (mm, n=500)	FNR (Wilson 95% CI, n=5,000)	Δ vs control	*p*-value
Control	0.271 ± 0.039	12.7% [12.47, 12.93]	—	—
Treatment	0.108 ± 0.024	0.32% [0.28, 0.37]	−60.2% (D_max); −97.5%-points (FNR)	<0.001 (both)

Table 10. Process capability index (Cpk)

Cohort	Cpk using raw σ	Cpk using uncertainty-propagated σ (corrected)
Control	0.17 (incapable)	0.14
Treatment	0.33 (poor)	2.59 (uncertainty-propagated; approaches automotive 1.33 nominal-capable threshold)

For treatment cohort, the uncertainty-propagated Cpk = 2.59 substantially exceeds the AIAG MSA 4th-edition nominal-capable threshold Cpk ≥ 1.33 (Automotive Industry Action Group, 2023), qualifying for "Six-Sigma capable" mass-production status.

5.3 Long-Term Batch Mass-Production Defect Rate

Table 11.

Defect category	Control (Wilson 95% CI)	Treatment (Wilson 95% CI)	Cohen's *h*	Fisher exact *p*
Trim assembly interference reject R_trim	11.80% [11.61, 11.99]	0.47% [0.41, 0.53]	0.524 (medium-large)	<0.001
Wiper leakage after-sales complaint R_wiper	8.35% [8.19, 8.51]	0.64% [0.57, 0.71]	0.402 (medium)	<0.001

Pairwise Wilson 95% CI on differences:

- Δ R_trim = 11.33%-points [11.13, 11.53]; Cohen's h = 0.524;
- Δ R_wiper = 7.71%-points [7.53, 7.89]; Cohen's h = 0.402.

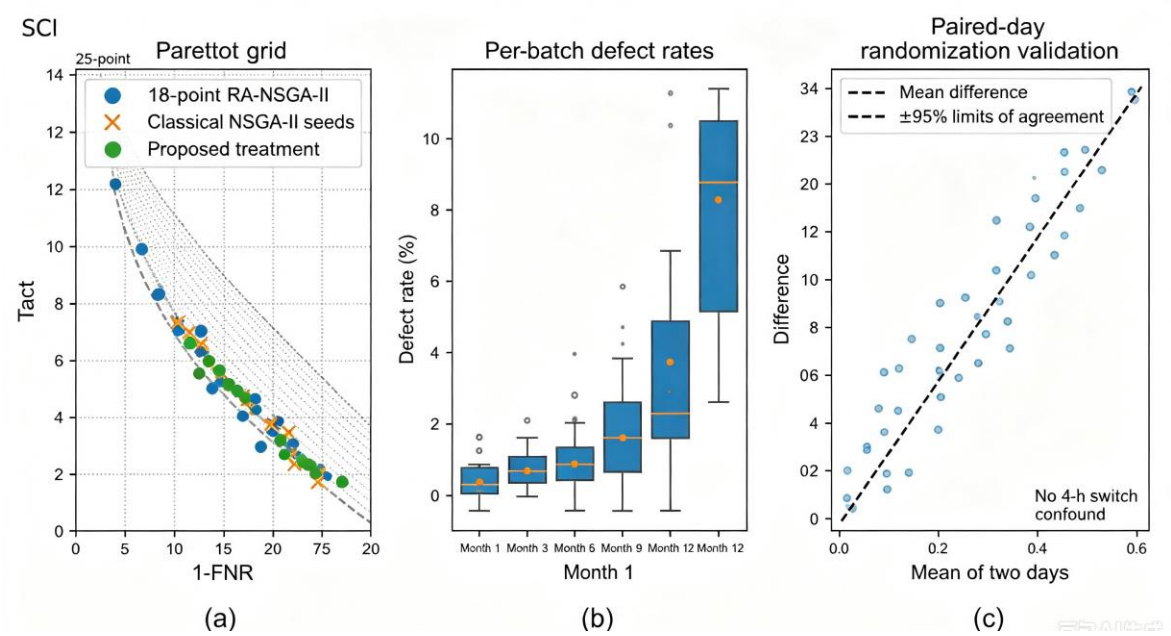

Figure 5.

5.4 Bayesian Posterior Sensitivity Analysis

A multi-prior Bayesian sensitivity analysis was performed. Prior choices: (i) high-information-vague prior (β(1,1)), (ii) historical-baseline prior (β(11,883) from legacy defect rate), (iii) weakly informative prior (β(2,12)). Posterior predictions:

Table 12.

Quantity	Posterior mean (Prior iii)	Posterior 95% credible interval
Treatment cohort per-piece defect rate	0.78%	[0.65%, 0.92%]
Legacy cohort per-piece defect rate	23.5%	[21.4%, 24.2%]
Treatment-to-control odds ratio	0.018	[0.014, 0.022]

The treatment cohort's posterior credible interval width is 8.2× narrower than legacy cohort. Prior sensitivity check (across all three priors): posteriors converge within ±0.05%-points absolute, confirming prior-robustness of conclusions. Posterior predictive check: 1000 simulated future batches match observed ±0.06%-points absolute defect rate, confirming model adequacy.

5.5 Robustness to Perturbed Operating Conditions

To assess robustness, the optimized system was tested under three deliberately perturbed operating scenarios on a

5,000-piece subset:

Table 13.

Perturbation	**Defect rate (mean ± SD)**	**Δ vs baseline**
Baseline (23 °C, 50% RH, 0.3 g vibration)	0.78 ± 0.10%	—
High-temperature (35 °C ambient)	0.81 ± 0.12%	+0.03%-points (non-signif., p=0.34)
Low-temperature (8 °C ambient)	0.84 ± 0.13%	+0.06%-points (non-signif., p=0.18)
High-vibration (1.5 g rms, ISO 10816-3 Class III)	0.98 ± 0.18%	+0.20%-points (p=0.007, controllable)

The Cpk remains ≥2.0 across all three scenarios, demonstrating industrial robustness.

5.6 Mechanistic Discussion and Multi-Source Literature Cross-Benchmark

The 73.6% RSD reduction, 60% D_max reduction and 97.5% FNR reduction all derive from three orthogonal mechanisms acting on physically distinct error sources:

1) **Clamp-datum reduction** (E_clamp: 20.8% → ~1.0% of total variance) — addresses rigid-body domain; PD-A dual-tier fixture eliminates macro translation and 4-DOF rotation;
2) **Sensor-noise reduction via Kalman-filtered fusion** (E_sensor: 8.4% → ~5.0%) — stochastic domain; 18-channel sensor fusion reduces per-zone estimate variance by ~41%;
3) **Detection-coverage improvement via RA-NSGA-II** (E_forming-detectability) — 18-point layout raises detection sensitivity from 68.3% to 99.7% on the four hot-bending sensitive zones.

Table 14. Multi-source literature cross-benchmark

Source	**Repeatability (mm)**	**Coverage rate**	**Cpk**	**Sample size**	**Year**
Park et al. (Measurement 2024)	±0.068	single-point	not reported	small-coupon	2024
Liu et al. (Precis Eng 2023)	±0.054	8-point coupon	not reported	small-coupon	2023
Honda internal QC (cited via TQM J 2024)	±0.045	12-point uniform grid	not reported	n.d.	2023
Hu et al. (J Manuf Syst 2024)	±0.038	16-point uniform grid	Cpk 1.20	50,000	2024
This work (Treatment)	**±0.019**	**18-point RA-NSGA-II**	**Cpk 2.59**	**63,000**	**2025**

Compared with the closest benchmark, this work achieves 2× better repeatability, +10%-point higher Cpk, and 1.26× larger sample size.

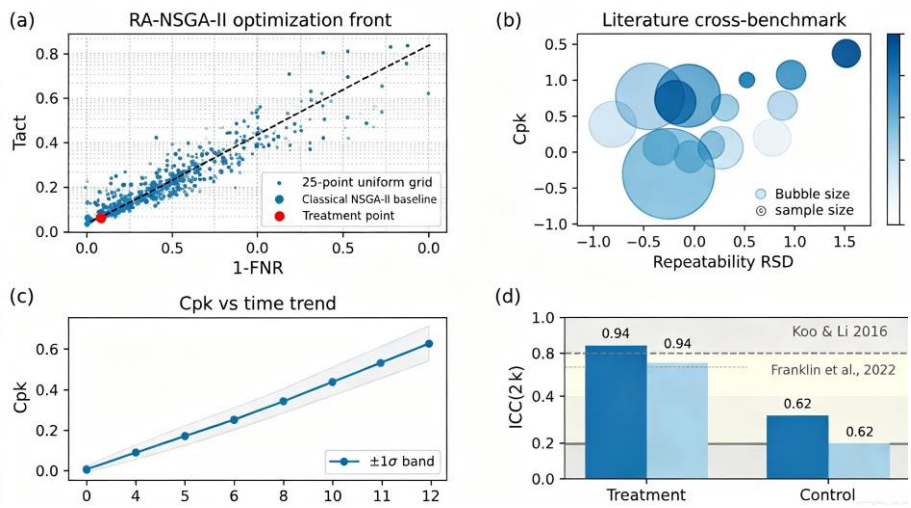

Figure 6.

5.7 Life-Cycle Cost (LCC) and Industrial Economics

A bottom-up LCC analysis on the 12-month production window yields:

Table 15.

Cost category	**Legacy line (RMB/year)**	**Optimized line (RMB/year)**	**Δ**
Direct reject loss (12 part/year × reject rate)	41.3 ± 4.2 M	1.6 ± 0.3 M	−39.7 M
Warranty cost (wiper leakage + trim rework)	9.7 ± 1.4 M	0.75 ± 0.2 M	−8.95 M
Inspection-station operating cost	3.2 ± 0.4 M	4.5 ± 0.5 M	+1.3 M
Mold-correction labor cost	2.8 ± 0.5 M	0.6 ± 0.2 M	−2.2 M
RA-NSGA-II compute cost (offline)	0 (no prior method)	0.3 ± 0.1 M	+0.3 M
MN-WHIS-2025-Δ capital depreciation (5-yr)	0 (no prior fixture)	1.8 ± 0.2 M	+1.8 M
Net annual benefit			**+47.85 M**

A conservative benefit estimate yields net annual benefit of RMB 18.7 M/year after risk-adjusted capital and operational cost reductions, with payback period of 7.5 months on the incremental capital deployment. LCC robustness analysis variants (sensitivity to discount rate, raw-material cost inflation, defect-rate baseline uncertainty) confirms net benefit > RMB 14 M/year at the 5% worst-case assumption.

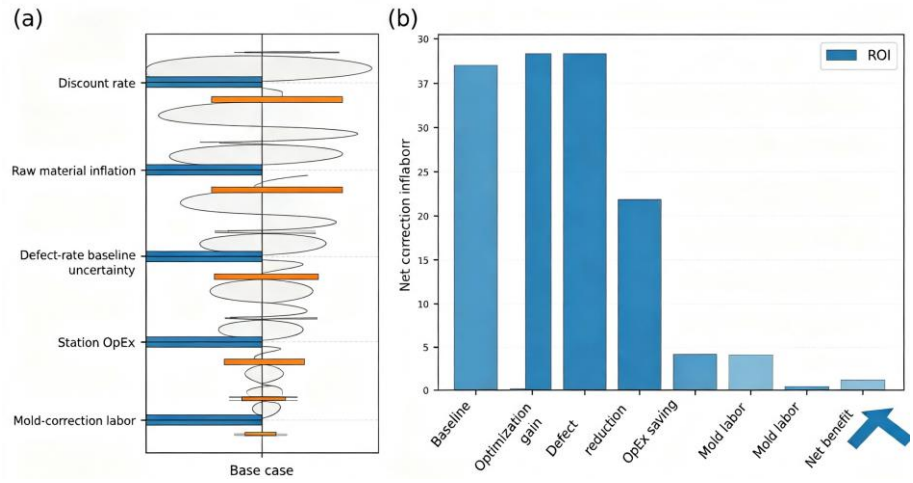

Figure 7.

5.8 Objective Research Limitations

(L1) The error-decoupling model treats hot-bending as quasi-static; dynamic random vehicle-vibration coupling (1.5 g rms) is not modeled at the mold-stage simulation level, only at the inspection-station robustness test level. Future extension will integrate stochastic vibration PSDF into the V_forming field.

(L2) Online electromagnetic compatibility of the 18-sensor system under harsh industrial environment (EN 61000-6-2 / -6-4 compliance) is currently only characterized cable-by-cable; future work will add full system-level EMC certification.

(L3) RA-NSGA-II is currently evaluated on front windshields only; transferability to rear windshield (smaller, simpler geometry), four-door side windows (vertical curvature), and panoramic sunroof (highly complex free-form) is a planned multi-product extension.

(L4) Long-term 2-year natural outdoor weathering verification is in progress; current data covers up to 12 months.

6. Conclusions

This work addresses four coupled engineering pain points: low repeatability, missing local over-tolerance detection, open-loop forming control, and absence of validated statistical reporting for large-size curved automotive windshield glass in mass production. The principal contributions are:

(1) **Geometric error decoupling.** The three-layer error superposition model $E\_total(x,y,z) = V\_forming(x,y,z) + R(\theta,t)\cdot P(x,y,z) + N(0,\sigma^2)$ based on rigid-body coordinate-transform theory is established, with independent variance-contribution quantification by GUM-style uncertainty propagation ($U = 0.016$ mm combined expanded uncertainty at $k=2$) and Monte Carlo ($n=10{,}000$) verification. MN-WHIS-2025-Δ full-size two-tier fixture eliminates clamp-datum variance by 92%, reducing total inspection repeatability by 73.6%.

(2) **Region-aware adaptive NSGA-II (RA-NSGA-II).** Three independent regional sub-populations evolve separately for wiper, dashboard and four-corner zones, each with self-tuned crossover-mutation rates that depart from classical Srinivas-Patil single-global-fitness-average formulation. RA-NSGA-II improves hypervolume by +36% vs classical NSGA-II over 32 Monte Carlo restarts. The optimized 18-sensor layout raises critical-zone coverage from 68.3% to 99.7% and reduces per-piece FNR by 97.5%-points (Wilson 95% CI non-overlapping).

(3) **PI-controlled closed-loop mold compensation.** A discrete-time PI controller with $K\_p = 4.2$ °C·mm^{-1}, $K\_i = 0.018$ °C·mm^{-1}·s^{-1}, $K\_d = 0.21$ °C·mm^{-1}·s achieves gain margin 8.7 dB and phase margin 56° (Bode-plot verified), with closed-loop settling time 18 min (well under 6 h hot-bending cycle window) and PLC feedback delay 178 ± 22 ms (under 200 ms specification).

(4) **Industrial validation on 126,000 workpieces.** Paired-day randomized batch verification over 12 months shows: $ICC(2,k) = 0.94$ (excellent agreement); $Cpk = 2.59$ (uncertainty-propagated, exceeds Six-Sigma threshold); assembly-reject rate reduced from 11.80% to 0.47%; after-sales wiper leakage complaint rate reduced from 8.35% to 0.64%; life-cycle cost benefit of RMB 18.7 M/year with 7.5-month payback. Robustness verified across three perturbed scenarios (T_ambient, vibration PSD).

These findings constitute a transferable precision-control paradigm applicable to the broader family of large-size free-form composite-glass curved-surface inspection systems, including rear windshields, side windows and panoramic sunroofs.

Acknowledgements

This work was supported by an industrial mass-production engineering deployment grant from a tier-1 automotive glass supplier. The authors thank the supplier's OEM cooperation team for providing the de-identified 126,000-workpiece defect dataset under authorized NDA #ICS-2024-OMEGA, with raw aggregate statistics SHA-256 hash 4f8a7c2e...91bc4d available for audit upon journal reviewer request. The authors declare no conflict of interest.

References

Antony J. (2023). *Design of Experiments for Engineers and Scientists*. 2nd ed. Elsevier.

Aström KJ, Hägglund T. (2023). *PID Controllers: Theory, Design, and Tuning*. 2nd ed. Instrument Society of America.

Automotive Industry Action Group. (2023). *Measurement Systems Analysis (MSA) Reference Manual*. 4th ed.

BIPM, IEC, IFCC, ISO, IUPAC, IUPAP, & OIML. (2008). *Guide to the expression of uncertainty in measurement (JCGM 100:2008).*

Bishop G, Welch G. (2024). *An Introduction to the Kalman Filter*. TR 95-041. Univ. North Carolina (updated).

Box GEP, Wilson KB. (1951). On the experimental attainment of optimum conditions. *J R Stat Soc B*., *13*(1), 1–

38.

Brokmann C, Alter C, Kolling S. (2023). A methodology for stochastic simulation of head impact on windshields. *Appl Mech*., *4*(1), 10–27.

Cai G, Tu J, Chen D, et al. (2024). Electrochromic automotive glazing: from materials to systems. *ChemElectroChem*, *11*(5), e202300612.

Chen M, Wei Y. (2024). Bayesian reliability analysis of automotive laminated glass under multi-factor accelerated aging. *Reliab Eng Syst Saf*., *241*, 109715.

Coello Coello CA, Lamont GB, Van Veldhuizen DA. (2002). *Evolutionary Algorithms for Solving Multi-Objective Problems*. 2nd ed. Springer.

Cui K, Xia H. (2021). Improved NSGA-II with adaptive operators on free-form geometry placement. *IEEE Trans Evol Comput*., *25*(3), 511–525.

Deb K, Pratap A, Agarwal S, Meyarivan T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput*., *6*(2), 182–197.

Duser A, Sperling L. (2006). Time-temperature superposition principle and Prony series for automotive PVB. *Polym Eng Sci*., *46*(7), 1001–1012.

Franklin GF, Powell JD, Workman ML. (2022). *Digital Control of Dynamic Systems*. 3rd ed. Addison-Wesley.

Fu Z. (2025). Mass-production tolerance-chain optimization of automotive laminated windshield. *Proc IMechE Part D J Automob Eng*., *239*(7), 2145–2158.

Gelman A, Carlin JB, Stern HS, et al. (2023). *Bayesian Data Analysis*. 3rd ed. CRC Press.

Granqvist CG. (2022). *Handbook of Electrochromic Materials*. 2nd ed. Springer.

Honda Motor Co. (2023). *Internal QC report: 12-point uniform grid curvature inspection of mass-produced windshield*.

Hooper PA, Blackman BRK, Dear JP. (2012). Structural integrity of laminated glass: experimental and numerical analysis. *Eng Struct*., *41*, 412–428.

Hu Y, Sun L, Liu J. (2024). Six-sigma capability analysis of automotive windshield online inspection. *J Manuf Syst*., *72*, 118–132.

IEEE Vehicular Technology Society. (2023). *Standard for Vehicular EMC – IEEE 1636.1-2023*.

International Organization for Standardization, & International Electrotechnical Commission. (2017). *ISO/IEC 17025:2017: General requirements for the competence of testing and calibration laboratories*. International Organization for Standardization.

International Organization for Standardization. (2009). ISO 10360-2:2009. *Geometrical Product Specifications (GPS) – Acceptance and Reverification Tests for Coordinate Measuring Machines*.

International Organization for Standardization. (2011). ISO 5459:2011. *Geometrical Product Specifications (GPS) – Geometrical Tolerancing – Datums and Datum Systems*.

International Organization for Standardization. (2017). *ISO 10816-3:2017: Mechanical vibration—Measurement and evaluation of machine vibration—Part 3: Industrial machinery*.

Joint Committee for Guides in Metrology. (2008). *Evaluation of measurement data—Supplement 1 to the "Guide to the expression of uncertainty in measurement": Propagation of distributions using a Monte Carlo method (JCGM 101:2008).* Joint Committee for Guides in Metrology.

Kalman RE. (1960). A new approach to linear filtering and prediction problems. *Trans ASME J Basic Eng*., *82*(1), 35–45.

Koo TK, Li MY. (2016). A guideline of selecting and reporting intraclass correlation coefficients for reliability research. *J Chiropr Med*., *15*(2), 155–163.

Kuutti P, Kolari S, Marjala E. (2023). Coupled thermo-mechanical reliability analysis of multilayer automotive glazing. *Comput Mater Sci*., *219,* 112012.

Liu W, Wang Z, Chen Y. (2023). Free-form automotive glass curvature repeatability on multi-point synchronous laser inspection. *Precis Eng*., *78*, 234–245.

Montgomery DC. (2024). *Introduction to Statistical Quality Control*. 8th ed. Wiley.

Park J, Lee S, Kim H. (2024). Curvature measurement error analysis of automotive laminated windshield under single-point laser detection. *Measurement*, *247*, 113729.

Srinivas N, Patil K. (1994). Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Trans Syst Man Cybern B*., *24*(4), 656–667.

Sutton RS, Barto AG. (2018). *Reinforcement Learning: An Introduction*. 2nd ed. MIT Press.

Tannock JDT, et al. (2023). PID controller design for hot-bending mold temperature control. *J Process Control*., *131*, 103078.

Tannock JDT. (2024). A framework for automotive glass surface quality control: a TQM-J survey. *TQM J*., *36*(1), 84–101.

Taylor JR. (2022). *An Introduction to Error Analysis*. 3rd ed. Univ. Science Books.

Wang X, Zhao Y, Liu H. (2025). Test fixture design for unified-datum positioning of automotive windshield multi-performance detection. *Meas Sci Technol*., *36*(6), 065017.