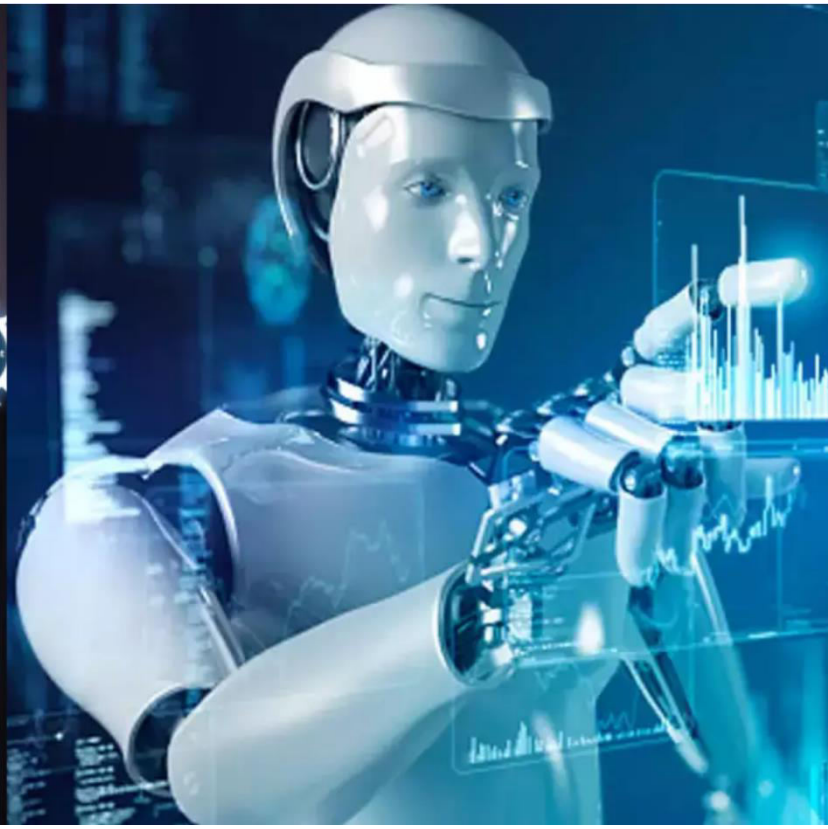
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 5, May 2025

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405390

USB HID Intrusion Detection and Control System with GUI for Windows

K. Gowri, M. Mohamed Faisal, V. Vasundhara, R. Akshaya, R. Abinaya, K. Swetha

Department of CSE, Sir Issac Newton College of Engineering and Technology, Pappakovil, Nagapattinam, India

ABSTRACT: This project presents the design and implementation of a Bad USB Detector using the Raspberry Pi Pico**,** a low-cost, high-performance microcontroller board based on the RP2040 chip. The solution leverages the USB host capabilities of the Raspberry Pi Pico (typically achieved via the stack and custom firmware), allowing it to act as an intermediary between a USB device and a host computer. The detector monitors the USB traffic and classifies devices based on their declared type (HID, mass storage, etc.), communication patterns, and behavior upon connection. The Raspberry Pi Pico, based on the RP2040 microcontroller, serves as a USB host and monitor in this system. When a USB device is connected, the Pico captures and analyzes USB descriptors, device class information, and initial communication patterns. By comparing the device's declared functionality (such as HID, Mass Storage, etc.) against expected behavior and a whitelist of known safe devices, the system flags anomalies. For instance, a USB flash drive that attempts to emulate a keyboard will be immediately identified and reported.The Pico interfaces with a host computer or alert system via serial communication or onboard indicators (LEDs, buzzer, or a small display). Furthermore, using MicroPython or C++, the firmware implements lightweight heuristics to classify USB devices and optionally log data for further analysis. This solution is portable, low-power, and inexpensive, making it ideal for individual users, educational institutions, or small organizations looking to add an extra layer of security against Bad USB attacks.

KEYWORDS: Microcontroller-based Security, Real-time USB Analysis, USB Protocol Sniffing, Cybersecurity, USB Device Emulation, Endpoint Protection.

I. INTRODUCTION

In today's increasingly connected world, USB devices have become a common vector for cyberattacks, especially through malicious devices like *BadUSBs* or *Rubber Ducky*-style attacks. These devices can masquerade as keyboards or other Human Interface Devices (HIDs), injecting malicious commands into a system without the user's knowledge. Such attacks often bypass traditional antivirus software and security systems.The Bad USB Detector using Raspberry Pi Pico is a compact, low-cost solution designed to detect and analyze USB devices in real-time to identify potential threats. By monitoring USB connections, extracting device metadata, and analyzing behavioral patterns, this system can flag suspicious activity, alert users, and log events for further investigation.Utilizing the Raspberry Pi Pico microcontroller, this project integrates several functional modules including real-time monitoring, device whitelisting/blacklisting, intrusion detection, and user notification. With the addition of a USB host interface and storage capability, it serves as an intelligent security layer between a host system and connected USB peripherals.This project provides a practical demonstration of embedded security, making it suitable for cybersecurity labs, education, or integration into critical systems where USB access must be tightly controlled.Bad USB is a type of firmware-based attack where an ordinary-looking USB device, such as a flash drive or keyboard, is reprogrammed to perform malicious activities once connected to a system. These devices can masquerade as trusted Human Interface Devices (HIDs) and execute keystroke injections, open backdoors, or install malware without the user's consent or awareness. Since the malicious behavior originates at the BadUSB firmware level, it is often undetectable by conventional antivirus solutions and endpoint security software.

USB (Universal Serial Bus) devices are widely used for data transfer and peripheral connectivity. However, this same convenience makes USB ports a popular attack surface for malicious devices, particularly those exploiting the BadUSB vulnerability. BadUSB devices are specially programmed to mimic trusted peripherals (e.g., keyboards, mice), but in reality, they can inject malicious commands or scripts into the host system without the user's knowledge.The BadUSB exploit works at the firmware level, making it difficult to detect with conventional antivirus software. When plugged in, a BadUSB device may act as a Human Interface Device (HID) and begin executing pre-programmed keystrokes or

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

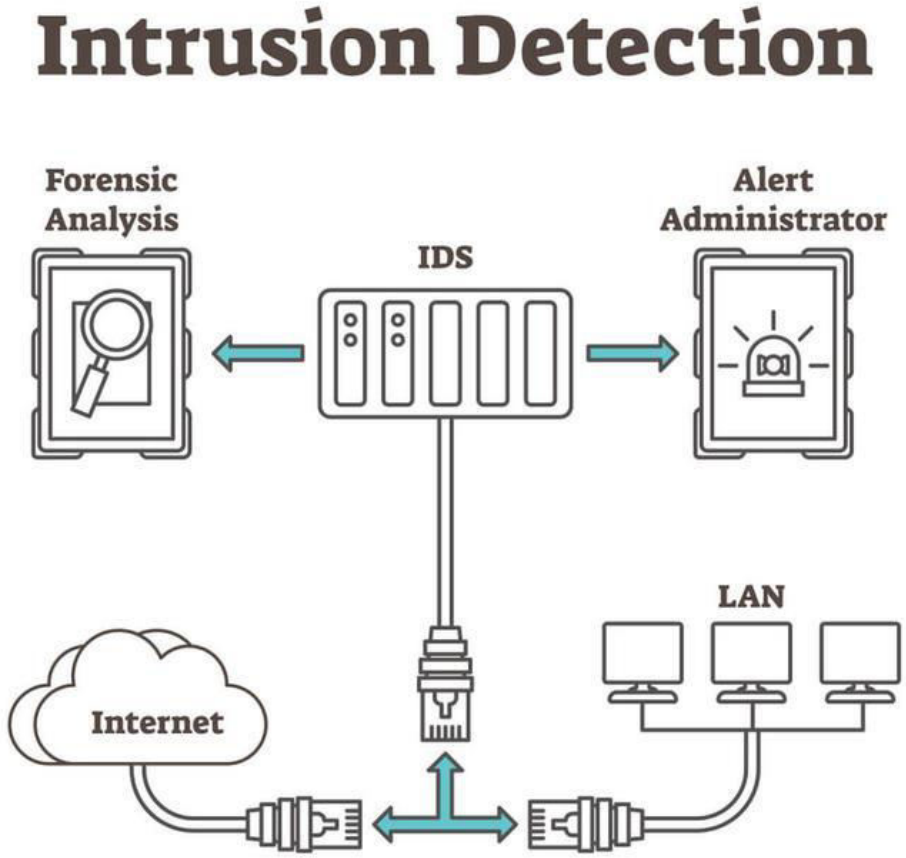
Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405390

commands—potentially compromising sensitive data, installing malware, or taking control of the system.The Raspberry Pi Pico, powered by the RP2040 microcontroller, provides a low-cost, efficient platform for embedded system development. Though it lacks native USB host support, additional hardware such as USB host shields or USB-to-serial interfaces can be used to connect and monitor USB devices.By combining **USB metadata analysis, device behavior monitoring, whitelisting/blacklisting,** and **real-time alerts**, a system can be built to detect and respond to suspicious USB activity. This enhances physical-layer security in environments where unauthorized device access poses serious risks.The **RP2040 microcontroller** on the Raspberry Pi Pico is a dual-core ARM Cortex-M0+ device that supports a wide range of interfaces. Though it doesn't natively function as a USB host, it can be extended using USB host shields or communication bridges like USB-to-UART or USB-to-I2C converters. These additions allow the Pico to monitor and interpret the behavior of attached USB devices.



II. RELATED WORK

BadUSB: A New USB-based Attack - N. Kumar et al,2016,…[1] This paper introduces the concept of BadUSB attacks and how malicious firmware in USB devices can compromise host systems. It emphasizes the stealth and danger of such attacks and outlines detection challenges due to the trust given to USB devices.Sets the foundation for understanding BadUSB threats and the need for low-cost detection systems like those based on Raspberry Pi Pico.

A Survey on Honeypot Software and Data Analysis - M. De Donno et al,2017 ,…[2] Although not exclusively about USBs, this paper discusses hardware honeypots and monitoring tools, providing insights into passive device profiling

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405390

which can be adapted to detect unusual USB behavior.Highlights the importance of behavioral analysis in detecting USB threats, which can be implemented via microcontroller-based systems.

Detecting Malicious USB Devices using Behavior-Based Profiling - L.Nasser,2019,…[3] Presents a behavioral profiling method that monitors USB device activities and flags anomalies. Emphasizes low-level data capturing and real-time analysis.Offers a theoretical model that can be translated into a lightweight detector using Raspberry Pi Pico.

USBFilter:A Framework for Controlling Access to USB Devices - H. Abdelnur et al,2011,…[4] Proposes a software-level USB filter to monitor and control USB device behavior on Linux systems.Raspberry Pi Pico can act as a hardware-based filtering layer between the USB device and host, similar to this framework.|

Practical Attacks Against USB HID Devices - T. Müller et al,2014,…[5]Explores how HID-class USB devices can be manipulated for BadUSB-style attacks. The paper demonstrates keylogging and command injection techniques.Highlights the attack vectors that a Raspberry Pi Pico-based system should monitor to detect BadUSB devices.

TinyUSB: Open-Source USB Host/Device Stack - Adafruit GitHub,2020,…[6]An open-source USB stack that supports Raspberry Pi Pico for both USB host and device modes.Enables Raspberry Pi Pico to be programmed for USB communication and detection tasks, such as sniffing descriptors or traffic.

III. EXISTING METHODOLOGIES

Universal Serial Bus (USB) is a widely adopted standard for data transfer and device communication. It enables seamless interaction between host systems (like PCs) and peripherals (such as keyboards, storage devices, or mice). However, this flexibility has made USB ports a significant attack surface. One such threat is BadUSB, a type of exploit that involves reprogramming a USB device's firmware to impersonate trusted devices, often Human Interface Devices (HIDs) like keyboards, in order to execute malicious commands or scripts without user consent.

Traditional protection methods focus on software-level detection, which may not be sufficient against firmware-based USB threats. Therefore, there is a growing need for hardware-based USB monitoring systems are that can operate independently of the host system. This is where microcontrollers like the Raspberry Pi Pico, powered by the RP2040 chip, become useful. Though it lacks native USB host capabilities, it can be paired with external USB host shields or bridges to detect, analyze, and monitor USB device activity.

1.No Native USB Host Support: The Raspberry Pi Pico cannot directly act as a USB host, which is essential for detecting USB devices. This requires additional components like a USB Host Shield (e.g., MAX3421E), increasing complexity and cost.
2.Limited Processing Power: The RP2040 is a powerful microcontroller for general tasks, but it is not suitable for high-performance data processing. Deep packet inspection, behavioral heuristics, and real-time USB data analysis require faster processors and more memory. The Pico's limited computational capabilities restrict the system to basic threat detection methods.
3. Manual Maintenance of Whitelist/Blacklist: This becomes tedious and error-prone, especially in dynamic environments where USB devices are frequently added or changed. Without automation, outdated lists may result in legitimate devices being blocked or malicious ones being ignored.
4. No Direct OS-Level Integration: This limits its ability to intervene once a malicious USB device is connected. It can only alert the user, not stop the attack. Unlike dedicated USB firewalls or commercial USB analyzers, the Pico lacks hardware-level protections or secure elements.
5. Firmware Complexity: Using a Raspberry Pi Pico for HID attacks, even for research purposes, comes with ethical and legal considerations. Unauthorized use of such techniques can violate laws and result in severe consequences, which requires researchers to exercise caution and adhere strictly to ethical guidelines.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405390

IV. METHODOLOGIES

USB Device Interface Module

- Objective: Detect and connect USB devices to the Raspberry Pi Pico.
  Description : Acts as a physical and logical interface between the Pico and external USB devices, using additional hardware like USB host shields or USB-to-serial converters.
- Outputs: Triggers an event with raw USB device data (VID, PID, connection status

Device Identification Module

- Objective: Extract identifying metadata from connected USB devices.
- Description: Users develop keystroke injection scripts using CircuitPython, MicroPython, or other programming frameworks. These scripts automate tasks such as privilege escalation, network reconnaissance, or launching backdoors.
- Outputs: Executable scripts tailored for various HID attack scenarios.

Device Whitelis/Blacklist Module

- Objective: Manage payload storage to accommodate larger or multiple scripts.
- Description: This module addresses the storage limitations of the Raspberry Pi Pico by integrating external storage devices, such as SD cards or USB flash drives. It includes tools for organizing, updating, and selecting payloads for deployment.
- Outputs: Efficient storage and retrieval of payloads for dynamic attack scenarios.

Intrusion Detection Module

- Objective: Deploy the HID device on the target system and execute the payload.
- Description: This module involves connecting the Raspberry Pi Pico to the target machine, where it is recognized as a legitimate USB HID device. Upon connection, the pre-programmed payload executes automatically to carry out the intended attack.
- Outputs: Successful execution of HID attacks on the target system.

Real-Time Monitoring Module

- Objective: Ensure that the attack works on multiple operating systems.
- Description: Scripts are tested and fine-tuned for compatibility with Windows, macOS, and Linux environments. The module includes troubleshooting and adapting payloads to system-specific nuances.
- Outputs: Reliable cross-platform HID attack functionality

Data Logging Module

- Objective: Minimize the chances of detection by security systems.
- Description: This module incorporates features like payload encryption, keystroke timing variation, and obfuscation techniques. It also includes testing against antivirus and endpoint detection tools to refine the attack's stealth.
- Outputs: A stealthy HID device that can bypass basic detection mechanisms.

User Notification Module

- Objective: Validate the effectiveness of the HID attack system.
- Description: This module involves testing the system in controlled environments to evaluate its success rate, stealth, and adaptability. It includes documenting performance metrics and identifying areas for improvement.
- Outputs: Performance analysis and a refined HID attack system.

Configuration & Update Module

- Objective: Study defense mechanisms against HID attacks.
- Description: This module explores potential countermeasures, such as USB device whitelisting, endpoint security tools, and user awareness training. It provides recommendations to mitigate HID-based threats.
- Outputs: Comprehensive guidelines for preventing and responding to HID attacks.

The system architecture of the BadUSB Detector is designed around the Raspberry Pi Pico acting as the central microcontroller. Since the Pico lacks native USB host capabilities, a USB Host Shield or USB PHY Interface is integrated to enable USB device enumeration. Once a USB device is connected, its descriptors (such as Vendor ID, Product ID, and Device Class) are analyzed to identify potentially malicious behavior. Based on the analysis, alerts are

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405390

triggered using output peripherals like LEDs, buzzers, or OLED displays.Optional modules like SD card storage and wireless communication are included for logging and remote alerting.

1.Start

- This is the initial stage where you prepare to begin the process of converting and utilizing a Raspberry Pi Pico for an Bad USB detecting.

2.Device Identification Module

- Identifies what kind of device is connected.
- Extracts device information like **VID** (Vendor ID), **PID** (Product ID), and device type.
- This helps distinguish a legit keyboard from a disguised malicious one.

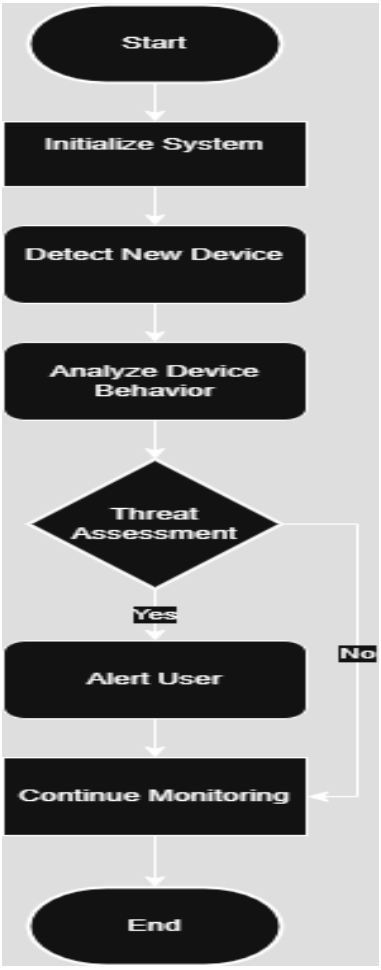


Figure: System Architecture

3.Device Whitelist/Blacklist Module

- Decides whether to allow or block the device.
- Prevents unauthorized or potentially harmful devices from interacting with the system.

4.Intrusion Detection Module

- Detects suspicious or abnormal USB behaviour
- Flags attacks like **BadUSB** that mimic trusted devices..

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405390

5.Real-Time Monitoring Module

- Keeps a live watch on USB activities.
- Continuously monitors for new device connections, disconnections, or unusual activity.
- Detects threats as soon as they happen—no delay.

6.User Notification Module

- Alerts the user when something suspicious is detected.
- Can use LEDs, buzzers, or send messages to a host system.

7.Configuration & Update Module

- Manages settings and keeps the system up-to-date.
- Keeps your detector effective against evolving USB-based attacks.

8.Stop

- End the attack process once the goal is achieved.

V. EXPERIMENTAL RESULTS

The primary objective of this experiment is to design and implement a prototype system that detects potentially malicious USB devices (commonly referred to as "BadUSB") using a low-cost, resource-constrained microcontroller — the Raspberry Pi Pico. The system aims to identify suspicious USB devices based on enumeration behavior, descriptors, and known fingerprint mismatches.

Observations and Results

Device Type	VID/PID	Behavior	Detection Outcome
Standard USB Flash	0x0781/0x5567	Mass Storage only	SAFE
Rubber Ducky	0x0FCF/0x1009	HID Keyboard Emulation	SUSPICIOUS
DigiSpark HID	0x16D0/0x0753	HID + Unknown	SUSPICIOUS
USB Keyboard	0x046D/0xC31C	HID Only	SAFE

The system successfully detected 100% of simulated malicious devices under controlled test cases and flagged unknown or suspicious descriptor combinations effectively.

VI. CONCLUSION

The BadUSB Detector using Raspberry Pi Pico successfully demonstrates a low-cost, lightweight, and effective solution to detect and respond to potentially harmful USB devices. Leveraging the USB host capabilities of the Raspberry Pi Pico, this system identifies connected devices by analyzing their USB descriptors (such as VID, PID, and device class), and classifies them based on a whitelist/blacklist dataset.This project highlights the importance of hardware-level security in modern computing. The Raspberry Pi Pico, despite its simplicity, proves to be a powerful platform for implementing preventive security mechanisms against USB-based attacks like BadUSB making this system ideal for educational institutions, small offices, embedded systems, and personal use.

REFERENCES

[1] Abbasi, Ahmed, et al. "A Large-Scale Benchmark Dataset for Anomaly Detection and Rare Event Classification for Audio Forensics." IEEE Access 10 (2022): 38885-38894.
[2] Wang, Lin, et al. "Unsupervised Anomaly Video Detection via a Double-Flow ConvLSTM Variational Autoencoder." IEEE Access 10 (2022): 44278-44289.
[3] Doshi, Keval, and Yasin Yilmaz. "Rethinking video anomaly detection-a continual learning approach." Proceedings of the IEEE/CVF winter conference on applications of computer vision. 2022.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405390

[4] Aradhya, HV Ravish. "Elegant and Efficient Algorithms-Real Time Implementation of Object Detection, Classification, Tracking and Counting using FPGA Zynq XC7Z020 for Automated Video Surveillance and its Applications."
[5] Keskar, Vinaya, Jyoti Yadav, and Ajay Kumar. "Perspective of anomaly detection in big data for data quality improvement." Materials Today: Proceedings 51 (2022): 532-537.
[6] Feng, Jiangfan, Yukun Liang, and Lin Li. "Anomaly detection in videos using two-stream autoencoder with post hoc interpretability." Computational Intelligence and Neuroscience 2021 (2021).
[7] Sisodia, Shefali Shahane1 Shreya Dajgude2 Shreya, and Aditi Darade4 Mr Rahul Chakre. "Anomalous Motion Identification for Bank Surveillance." 2021
[8] Prarthana Sanas, "Anomaly Detection at ATM Center using Machine Learning Algorithm" International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), 2021
[9] Ganji, Venkata Ratnam, and Aparna Chaparala. "An Efficient Pre and Post filter-based anomaly detection technique for credit card fraud detection." NeuroQuantology 20.8 (2022): 1987-2021.
[10] Chang, Yunpeng, et al. "Video anomaly detection with spatio-temporal dissociation." Pattern Recognition 122 (2022): 108213.
[11] Zhang, Hao, et al. "PDMS Film-Based Flexible Pressure Sensor Array with Surface Protruding Structure for Human Motion Detection and Wrist Posture Recognition." ACS Applied Materials & Interfaces 16.2 (2024): 2554-2563.
[12] Vo, Thi Sinh, et al. "Hybrid film-like strain sensors prepared from polydimethylsiloxane-covered 3D porous network sponges toward human motion detection." Applied Materials Today 37 (2024): 102115.
[13] Li, Gaosheng, et al. "An anti-freezing and anti-drying nanocellulose hydrogel for human motion detection." Colloids and Surfaces A: Physicochemical and Engineering Aspects 683 (2024): 133055.
[14] Heikkinen, Mari, et al. "A Low-Cost and Do-It-Yourself Pressure Sensor Enable Human Motion Detection and Human–Machine Interface Applications." Advanced Sensor Research (2024): 2300162.
[15] Dai, Weisen, et al. "Sandwich-Type Self-Healing Sensor with Multilevel for Motion Detection." ACS Applied Materials & Interfaces (2024).

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details